

BCP
User's Guide

Order No. AA-N269B-TK

May 1983

This document contains detailed instructions on installing and using the BCP software product on the RSX-11M and VAX/VMS operating systems.

REVISION/UPDATE INFORMATION: This is the second printing.

SOFTWARE VERSION: RSX-11M Version
4.0 or later
VAX/VMS Version
3.2 or later

Prepared by
Computer Special Systems

Digital Equipment Corporation • Nashua, NH 03062

1st Printing, October 1982

2nd Printing, May 1983

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may only be used or copied in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by DIGITAL or its affiliated companies.

Copyright © 1983 by Digital Equipment Corporation
All Rights Reserved

Printed in U.S.A.

The postage-prepaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist us in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

digital™

DATATRIEVE

DEC

DECmate

DECnet

DECset

DECsystem-10

DECSYSTEM-20

DECTape

DECUS

DECwriter

DIBOL

MASSBUS

PDP

P/OS

Professional

Rainbow

RSTS

RSX

UNIBUS

VAX

VMS

VT

Work Processor

To order additional copies of this document, contact the Software Distribution Center, Digital Equipment Corporation, Northboro, Massachusetts 01532.

CONTENTS

Page

PREFACE

PART I CREATE (INTERACTIVE GRAPHICS PROGRAM) USER'S GUIDE

CHAPTER 1 INTRODUCTION

1.1	GENERAL DESCRIPTION	1-1
1.2	OVERVIEW	1-1
1.3	BCP SOFTWARE CONCEPTS	1-2
1.4	BCP SOFTWARE COORDINATE SYSTEM	1-6
1.5	GRAPHIC OUTPUT ON THE LXY PRINTER/PLOTTER	1-7
1.6	BCPLIB GRAPHICS PLACEMENT	1-8
1.6.1	Insert Forms on the PLOT PAGE	1-8
1.6.2	Inserting Graphic Figures on the PLOT PAGE	1-8

CHAPTER 2 GUIDE TO USING THE CREATE PROGRAM

2.1	INTRODUCTION	2-1
2.2	ACCESSING AND USING CREATE	2-1
2.3	ENTERING CREATE COMMANDS	2-2
2.3.1	Command Formats	2-2
2.3.2	Parameters	2-3
2.3.3	Options	2-3
2.3.4	Data Entry	2-3
2.4	GUIDELINES FOR USING CREATE COMMANDS	2-4
2.5	CREATE POINTERS	2-9
2.5.1	What to Do if an Error Message Is Received Following a CREATE Command Entry	2-9
2.5.2	How to Generate a Multipage Plot	2-10
2.5.3	If the REQUEST BUFFER Is Filled	2-10
2.5.4	To Erase Previously Entered Commands	2-11
2.5.5	To Remove a Fixed Form	2-11
2.5.6	To Get Plot Positioning and Other Useful Information	2-11
2.5.7	To Request a Summary of CREATE Commands	2-11

APPENDIX A CREATE LISTINGS

A.1	BCODE	A-1
A.2	BLOCK	A-3
A.3	DEFFRM	A-5
A.4	DEFPNT	A-7
A.5	END	A-8
A.6	ENDFRM	A-9
A.7	ERASE	A-10
A.8	EXIT	A-12

CONTENTS (Cont)

		Page
A.9	GETERM	A-13
A.10	HDASH	A-15
A.11	HELP	A-17
A.12	HLINE	A-18
A.13	HTAB	A-20
A.14	MOVPTR	A-21
A.15	MOVREF	A-22
A.16	NEWPLT	A-23
A.17	PAGE	A-24
A.18	PROCES	A-25
A.19	PUTERM	A-26
A.20	REPEAT	A-28
A.21	SAVFERM	A-30
A.22	SETFIG	A-31
A.23	SETPAG	A-33
A.24	SETTAB	A-35
A.25	SHOW	A-36
A.26	TEXT	A-38
A.27	VDASH	A-40
A.28	VLINE	A-42
A.29	VTAB	A-44

APPENDIX B GLOSSARY

APPENDIX C ERROR MESSAGES

C.1	WARNINGS	C-1
C.2	ERROR	C-1

FIGURES

Figure No.	Title	Page
1-1	RSX-11M BCP System	1-3
1-2	VAX-11 BCP System	1-4
1-3	FORMS	1-9
1-4	Graphics Insertion	1-10

CONTENTS

Page

PART II BCPLIB (GRAPHICS LIBRARY) USER'S GUIDE

CHAPTER 1 INTRODUCTION

1.1	GENERAL DESCRIPTION	1-1
1.2	OVERVIEW.....	1-1
1.3	BCP SOFTWARE CONCEPTS	1-2
1.4	BCP SOFTWARE COORDINATE SYSTEM	1-6
1.5	GRAPHIC OUTPUT ON THE LXY PRINTER/PLOTTER	1-7
1.6	BCPLIB GRAPHICS PLACEMENT	1-8
1.6.1	Insert Forms on the PLOT PAGE	1-8
1.6.2	Inserting Graphic Figures on the PLOT PAGE	1-8

CHAPTER 2 BCP UTILITY LIBRARY (BCPLIB) OVERVIEW

2.1	INTRODUCTION	2-1
2.2	USING THE BCPLIB ROUTINES	2-2
2.2.1	Status Arguments	2-4
2.2.2	Format of Graphic Figure Routines	2-4
2.3	GUIDELINES FOR USING BCPLIB TO GENERATE PLOTS	2-5
2.4	COMPILING AND LINKING AN APPLICATION PROGRAM ..	2-10

CHAPTER 3 RSX-11M INSTALLATION NOTES

3.1	SYSTEM REQUIREMENTS	3-1
3.2	SYSTEM INSTALLATION	3-1

CHAPTER 4 VAX/VMS INSTALLATION NOTES

4.1	SYSTEM REQUIREMENTS	4-1
4.2	SYSTEM INSTALLATION	4-1

APPENDIX A BCPLIB SUBROUTINES

A.1	BCODE	A-1
A.2	BLOCK	A-3
A.3	DEFFRM	A-5
A.4	DEFLBL	A-7
A.5	ENDFRM	A-8
A.6	ERASE	A-9
A.7	EXAM	A-10
A.8	FINISH	A-12
A.9	GETFRM	A-13
A.10	GETLBL	A-15

CONTENTS (Cont)

		Page
A.11	HDASH	A-16
A.12	HLINE	A-18
A.13	INIT	A-20
A.14	MOVPTR	A-21
A.15	MOVREF	A-22
A.16	PAGE	A-23
A.17	PROCES	A-24
A.18	REPEAT	A-25
A.19	PUTFRM	A-26
A.20	SAVFRM	A-27
A.21	SETFIG	A-28
A.22	SETPAG	A-29
A.23	SETTAB	A-30
A.24	TAB	A-31
A.25	TEXT	A-32
A.26	VDASH	A-33
A.27	VLINE	A-34

APPENDIX B GLOSSARY

APPENDIX C ERROR MESSAGES

C.1	WARNINGS	C-1
C.2	ERROR	C-1

APPENDIX D BCP VERIFICATION

FIGURES

Figure No.	Title	Page
1-1	RSX-11M BCP System	1-3
1-2	VAX-11 BCP System	1-4
1-3	FORMS	1-9
1-4	Graphics Insertion	1-10
2-1	Overlay Guide	2-3
D-1	DEMO PLOT	D-3

MANUAL OBJECTIVES AND READER ASSUMPTIONS

Since this user's guide has two objectives, it is divided into two parts. Each part is a complete manual in itself.

Part I (CREATE USER'S GUIDE) contains the operation and command elements of the BCP interactive graphics program CREATE.

Part II (BCPLIB USER'S GUIDE) describes the BCP graphic subroutine library (BCPLIB) modules and how to use these modules to integrate graphic output into the user's specific application program.

INTENDED AUDIENCE

Part I (CREATE USER'S GUIDE) is intended for the user who has a basic understanding of operating computer programs from a terminal.

Part II (BCPLIB USER'S GUIDE) is intended for the user who is a computer programmer or knows how to write computer software.

This manual provides instruction in the use of BCP software. It does not provide information on the FORTRAN language, system sources, or any DIGITAL operating system. It also assumes familiarity with the following.

- For RSX-11M BCP:
 - RSX-11M Operating System;
 - RSX-11M MCR Operations Manual, AA-L678A-TC; and
 - PDP-11 FORTRAN-77 Language.
- For VAX-11 BCP:
 - VAX/VMS Operating System;
 - VAX/VMS Command Language User's Guide, AA-D023C-TE; and
 - VAX-11 FORTRAN Language.

DOCUMENT STRUCTURE

PART I

Chapter 1 consists of general information concerning the BCP software and introduces basic information needed to understand the BCP software and hence, the CREATE program.

Chapter 2 describes how to use the CREATE program.

Appendix A lists the CREATE commands in alphabetical order and describes each command in detail.

Appendix B is a glossary of terminology used in describing the BCP software in this document.

Appendix C is a list of error messages that could be returned by the BCP software.

PART II

Chapter 1 consists of general information concerning the BCP software and introduces basic information needed to understand the BCP software and hence, the BCPLIB subroutine library.

Chapter 2 describes how to use the BCPLIB subroutine library.

Chapter 3 describes the RSX-11M BCP software installation procedure. This includes both CREATE and BCPLIB installation.

Chapter 4 describes the VAX/VMS BCP software installation procedure. This includes both CREATE and BCPLIB installation.

Appendix A lists the BCPLIB subroutines in alphabetical order and describes each subroutine in detail.

Appendix B is a glossary of terminology used throughout this manual in describing the BCP software.

Appendix C is a list of error returns from the BCPLIB subroutines with a description of each.

Appendix D consists of the BCP software (CREATE and BCPLIB) installation verification procedures.

PART I

CREATE (INTERACTIVE GRAPHICS PROGRAM) USER'S GUIDE

1.1 GENERAL DESCRIPTION

The Barcode and Block Character Plotter (BCP) software package generates both barcode and block character printing on an LXY PRINTER/PLOTTER. Using an interactive user program and a library of graphic routines, the BCP software package can produce:

- CODE 39 barcode,
- Block characters,
- Horizontal and vertical lines,
- Horizontal and vertical bars (thick lines),
- Horizontal and vertical dash lines, and
- 0.1 inch text characters.

Typical uses include the creation of:

- Shipping and identification labels,
- Forms,
- Signs, and
- Inventory and accounting documents.

1.2 OVERVIEW

The BCP software package is made up of two components, each designed for a different user.

- CREATE is an interactive program for the nontechnical user. CREATE is built on the foundation provided by BCPLIB and gives the user an on-line graphics generation capability for use in the production environment.
- BCPLIB (BCP Graphic Utility Library) is available for the technical user or programmer experienced in writing computer software. BCPLIB is a library of software building blocks that serves as a foundation for the graphics applications. It provides graphic facilities for both user application software and the BCP CREATE Program.

There are two front-ends (or ways) into the BCP software (Figures 1-1 and 1-2). The front-ends are either through the:

- Interactive CREATE program, or
- Optional user application program.

Both of these programs are built upon BCPLIB, with each accepting commands from a system's terminal. These front-ends pass the command information on to BCPLIB. This command information is called the GRAPHIC REQUEST. GRAPHIC REQUESTS are processed by BCPLIB in two phases due to physical limitations placed on the program (such as LXY PRINTER/PLOTTER paper movement and computer memory).

Phase 1

During this phase, the GRAPHIC REQUESTS are accepted, evaluated, and stored (buffered) into a special area of the program's space in computer memory called the REQUEST BUFFER.

Phase 2

Phase 2 is entered as a result of a special request (or CREATE command) such as the PROCES request. During this phase, the requests in the buffer are processed into plottable information, which is then written or temporarily stored in the GRAPHICS FILE, and is later transferred to the LXY PRINTER/PLOTTER.

Phases 1 and 2 can be repeated alternately, until an entire plot (which may be more than one page of output) is completed.

Once the plot is completed, the contents of the GRAPHICS FILE can be transferred directly to the LXY PRINTER/PLOTTER via the following methods.

- For RSX-11M -- Use the MCR utility PIP (or PRINT facility).
- For VAX/VMS -- Use the PRINT/NOFEED command.

1.3 BCP SOFTWARE CONCEPTS

This section explains the concepts that must be understood in order to use the BCP software. They are:

- PLOT PAGE,
- SECTION,
- FORMS, and
- LABEL POINTS.

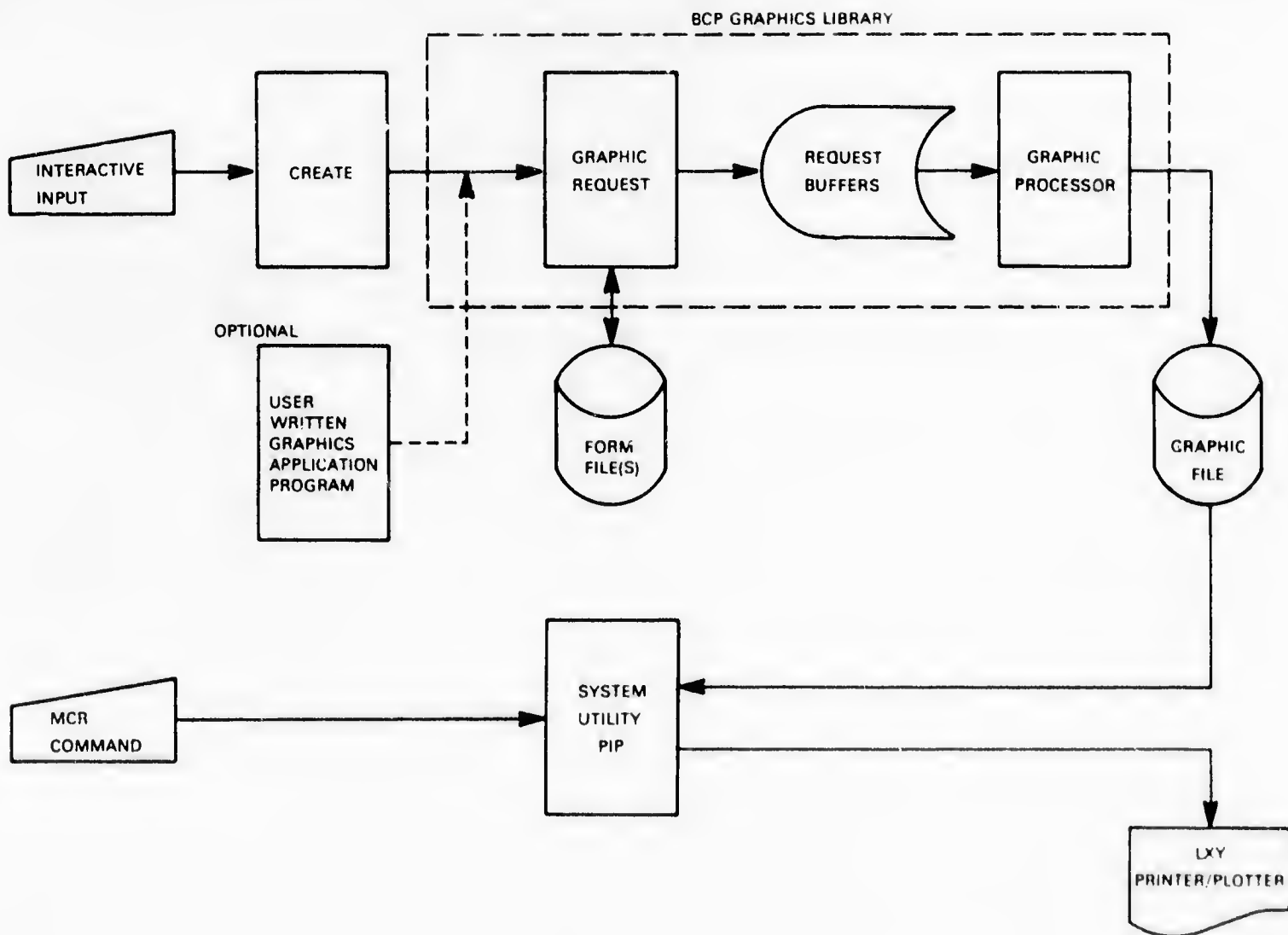


Figure 1-1 RSX-11M BCP System

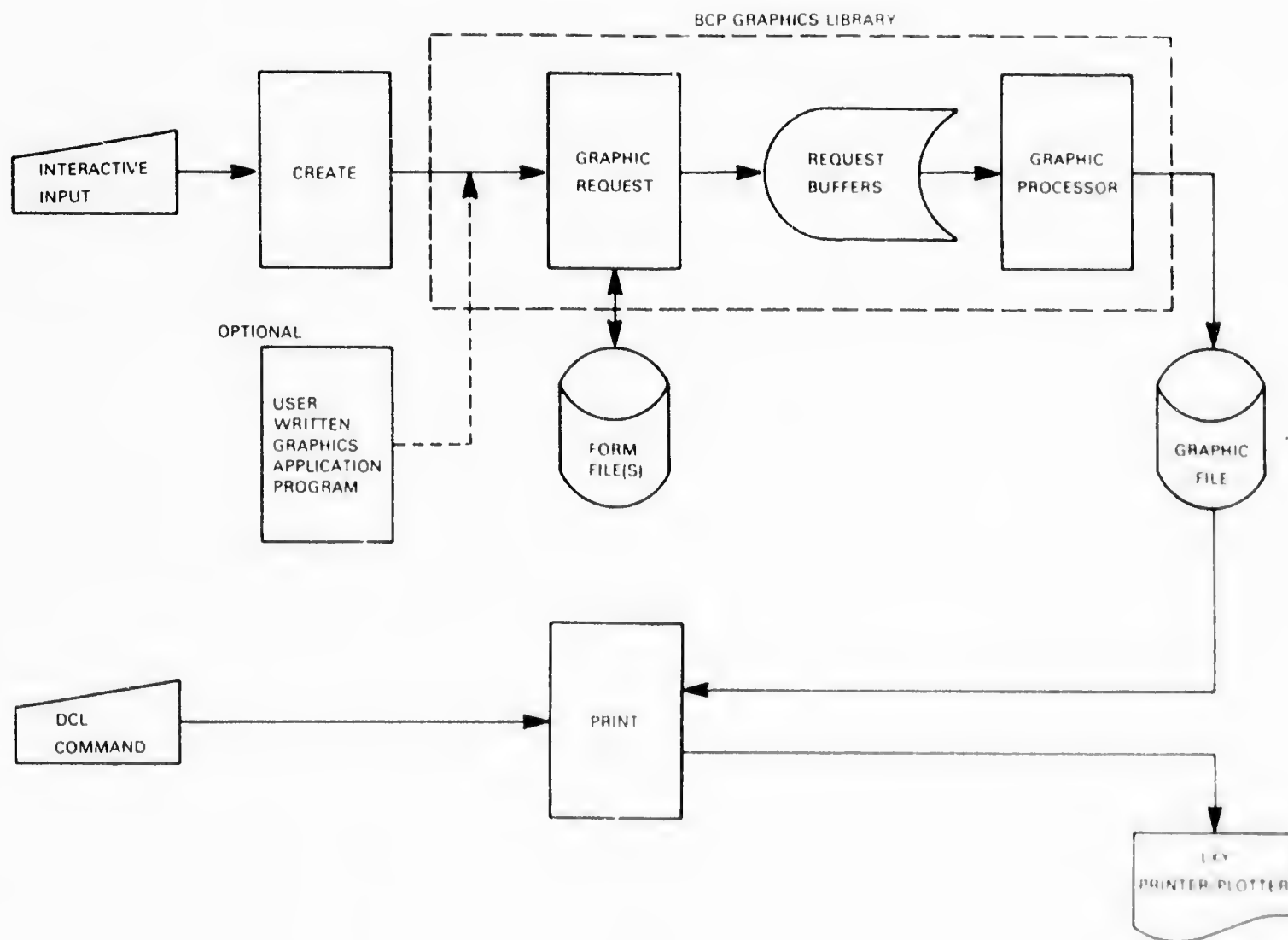


Figure 1-2 VAX-11 BCP System

PLOT PAGE

The PLOT PAGE is the basic unit of plot output. It is an internal software equivalent of the LXY PRINTER/PLOTTER page. Since the PRINTER/PLOTTER paper can be of different lengths or widths, the PAGE boundaries can also be adjusted internally to match the PRINTER/PLOTTER PAGE SIZE. When GRAPHIC REQUESTS are received by the software, the dimensions of the figures are checked to see if they fit within the page boundaries. GRAPHIC (or whatever) information can be generated for more than one page, but each page is treated individually and terminated by a "TOP-OF-FORM" paper movement. The "TOP-OF-FORM" movement positions the paper to the top of the next physical page.

SECTION

If a lot of GRAPHIC REQUESTS are required to generate a plot, the REQUEST BUFFER could overflow before the output page is complete. To prevent this, Phase 2 of the GRAPHIC REQUEST may be triggered at any time and the requests received up to that time are processed. The area of the page covered by the graphic information in the REQUEST BUFFERS determines the page SECTION. Once a SECTION is processed, no more graphic input is accepted that has any part falling within the range of that SECTION. Following the output of that SECTION, a new SECTION is defined by updating the SECTION origin position to the start of the unused part of the page. The REQUEST BUFFERS have a capacity that can be exceeded by the number of GRAPHIC REQUESTS, graphic data should be entered in a top-to-bottom sequence. However, for most plots, the REQUEST BUFFER is adequate to store the graphics for the entire page. SECTIONS are defined and written as the result of the:

- PROCES request,
- END request, and
- REPEAT request.

FORMS

The BCP FORMS facility provides a means for the user to generate the fixed portion of a graphic output (such as labels and business forms) and store it for later use. This means that the fixed part of a graphic output must be generated only once. When producing the final output, the user can merge the fixed portion with the variable part of the graphic output. FORMS can be created during the Phase 1 process of the GRAPHIC REQUEST. Graphic requests defining the FORM are processed in the same way as a normal output. Phase 2 processing is bypassed and on the completion of the FORM, a special request is called to reformat the contents of the REQUEST BUFFERS and write the requests on the FORMS file.

Once a FORM is created, it can be repeatedly recalled during software production runs by referencing its name. FORMS are called in a similar manner to standard graphic figures such as barcodes. When a FORM is called, the FORM base is aligned with the REFERENCE ORIGIN (Section 1.4) position on input. The FORM data is read from the FORM file, reformatted, and stored in the REQUEST BUFFER like any standard figure. The variable portion of the FORM can be inserted at this time by making additional GRAPHIC REQUESTS.

A FORM can be used to define another, more complex FORM. A FORM can also be fixed into memory for repeated use. Not only does this save I/O activity on the FORM file, but a single image of the FORMS graphics may be stored for any number of different positions of the FORM on the plot page, thus saving REQUEST BUFFER space.

LABEL POINTS

LABEL POINTS are used as an aid in the positioning of the variable portion of FORMS during production runs. LABEL POINTS are a set of up to 12 predefined positions on the FORM which are selected by number (1 through 12) to position the variable graphics. The X and Y values associated with each LABEL POINT are offset (or incremental) distances from the FORM base. The FORM base is aligned with the REFERENCE ORIGIN (Section 1.4) when the FORM is requested and the LABEL POINTS then become GRAPHIC POINTER (Section 1.4) positions on the PLOT PAGE. By requesting a LABEL POINT by number, the user causes the GRAPHIC POINTER to be moved to the position indicated (by the LABEL POINT) and additional graphic input can then be added.

1.4 BCP SOFTWARE COORDINATE SYSTEM

To aid in the placement of the graphic figures on the PLOT PAGE (see Figures 1-3 and 1-4), the BCP software provides a coordinate system that is defined as:

- PAGE ORIGIN,
- SECTION ORIGIN,
- REFERENCE ORIGIN, and
- GRAPHIC POINTER.

PAGE ORIGIN

Upper left-hand corner of the physical line printer paper. This is plot (or absolute) position (0,0).

The ABSCISSA Y is defined as being the vertical direction starting from the PAGE ORIGIN, increasing in value to the bottom of the page. Thus, the positive X direction corresponds with the paper movement and maintains a right-handed coordinate system. The resolution is 0.1 inch.

The ORDINATE Y is defined as being the horizontal direction starting from the PAGE ORIGIN, increasing in value to the right-hand side of the page. The resolution is 0.1 inch.

SECTION ORIGIN

This is the starting position of the current page section. The Y coordinate is always 0 and its X coordinate (XS) is an offset distance from the PAGE origin. The SECTION ORIGIN affects the user by becoming the lower limit for the X coordinate on his GRAPHIC REQUESTS. Initially, the SECTION ORIGIN starts at the PAGE origin which is 0,0. As sections are processed, it advances by the section height to define the start of the next section.

REFERENCE ORIGIN

This is a moveable coordinate system within the PAGE. Graphic figures are placed on the PAGE with respect to the REFERENCE ORIGIN (Xr,Yr). The X and Y position for the figures are offset distances from this point. The REFERENCE ORIGIN is moved by specifying new values for the Xr and Yr offsets from the PAGE origin. The REFERENCE ORIGIN is a mechanism to provide position independence (with respect to the PAGE) for the graphic figures on the PAGE. This is an important mechanism in processing FORMS and REPEAT requests.

The Xr value of the REFERENCE ORIGIN must not be smaller than the SECTION origin. As the PAGE sections are processed, the REFERENCE ORIGIN is advanced automatically with the SECTION origin.

GRAPHIC POINTER

Current position for graphic data entry is expressed as an offset Xc and Yc from the REFERENCE ORIGIN. Neither Xc nor Yc may take on negative values (for example, cannot be smaller than the REFERENCE ORIGIN).

1.5 GRAPHIC OUTPUT ON THE LXY PRINTER/PLOTTER

The LXY PRINTER/PLOTTERS place physical limitations on the BCP software operation. The PRINTER/PLOTTERS can only accept plottable data (raster data) one raster line (a line thickness of one dot, 0.0139 inch) at a time in a top-of-page to bottom-of-page and page-by-page sequence.

The LXY PRINTER/PLOTTER paper moves only in a forward direction, so that all graphic output must be coordinated with the printing movement of the paper. The plot data must be sorted and written on the PRINTER/PLOTTER one raster line at a time in a sequence acceptable to the PRINTER/PLOTTER. This places restrictions on the sequence of GRAPHIC REQUESTS entries by the user. As mentioned in Section 1.3, the BCP software sorts the stored requests according to physical position on the PAGE. Also, the BCP software can only accept graphic input in order according to PAGE. The restrictions to the order of graphic input on the PAGE depends on the number of requests and PAGE SECTIONS.

1.6 BCPLIB GRAPHICS PLACEMENT

1.6.1 Inserting Forms on the PLOT PAGE

FORMS are inserted on the PLOT PAGE at the current REFERENCE origin (Figure 1-3). The REFERENCE ORIGIN coincides with the FORM BASE. The FORM graphics are drawn from this point down the page (an increasing value of X) until the FORM length (FLIMX) is reached. In the same way, the FORM is drawn across the page until the FORM (FLIMY) width is reached.

The FORM LABEL POINTS, being offset values from the FORM BASE, now become GRAPHIC POINTER values.

The REFERENCE origin is moved in one of several ways. It may be moved:

- As the result of entering the TAB request (for example, VTAB or HTAB commands for the CREATE user or a call to the TAB subroutine for the BCPLIB user),
- As the result of entering the MOVREF request,
- As the result of entering the GETFORM and PUTFORM requests. These requests contain X and Y parameters to reposition the REFERENCE origin prior to the actual form insertion, or
- As the result of action taken at the completion of the PAGE, PRINT, and REPEAT requests. The REFERENCE ORIGIN is moved to the new PAGE or SECTION origin according to the request.

1.6.2 Inserting Graphic Figures on the PLOT PAGE

NOTE

Sufficient space (in both height and width) must be made available for the graphic figure that is to be drawn. The GRAPHIC POINTER position (Xc, Yc) must be chosen so that enough space in height and width is available to draw the graphic figure.

All graphic figures are inserted into the PLOT PAGE at the current GRAPHIC POINTER position. Figure 1-4 illustrates how the figure is placed using the block Character A as an example. The GRAPHIC POINTER (Xc, Yc) is assumed to be on the lower left corner of the figure, which is then drawn upward on the page (a decreasing X direction) until reaching the figure height and to the right by its width (SIZE).

The new position of the GRAPHIC POINTER is then set to the lower right corner figure plus an intercharacter space (Xce, Yce). The user should be cautioned that before placing the figure on the PLOT PAGE, the X values for the GRAPHIC POINTER position (Xc, Yc) must be at least as great as the figure height (SIZE).

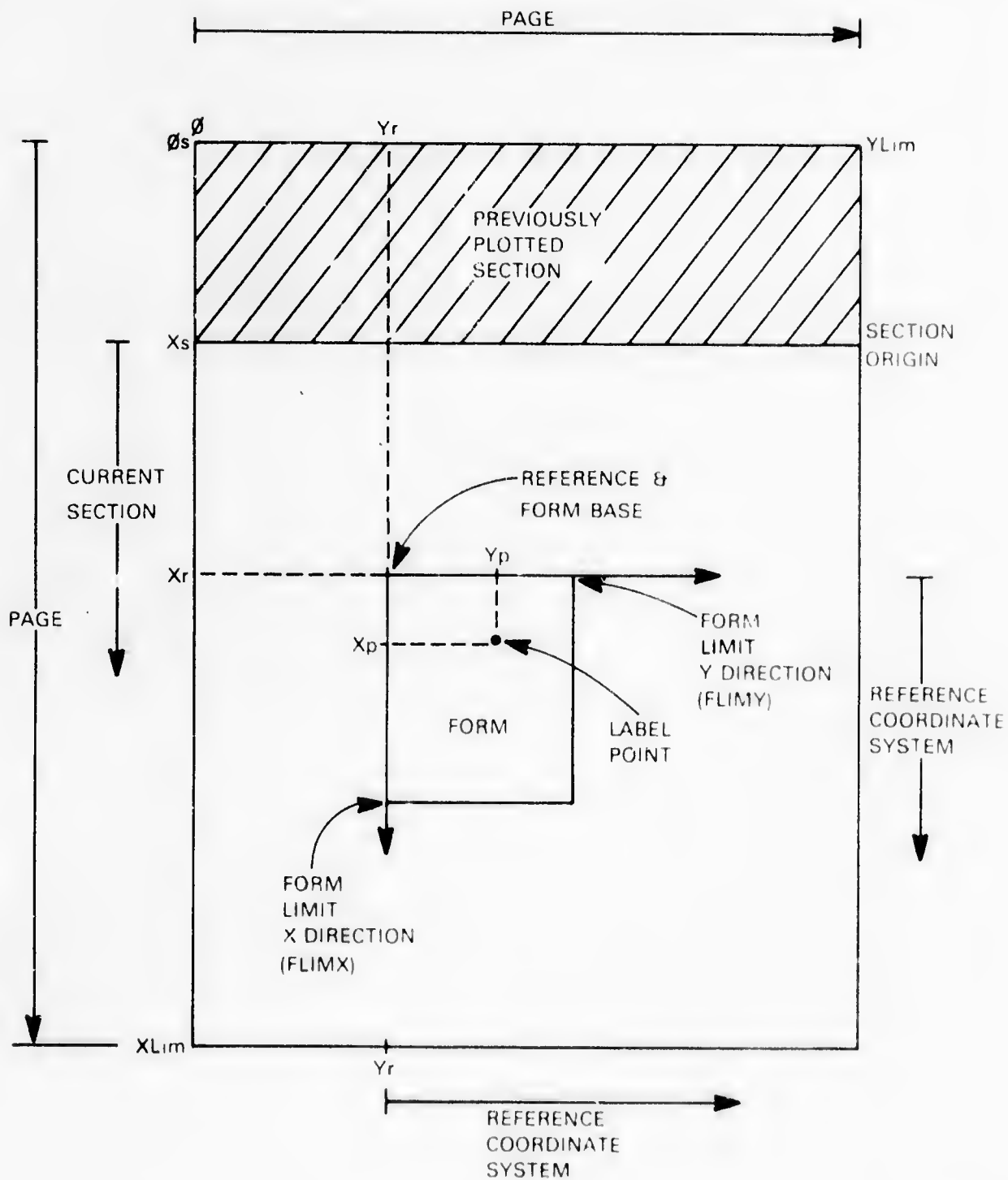


Figure 1-3 FORMS

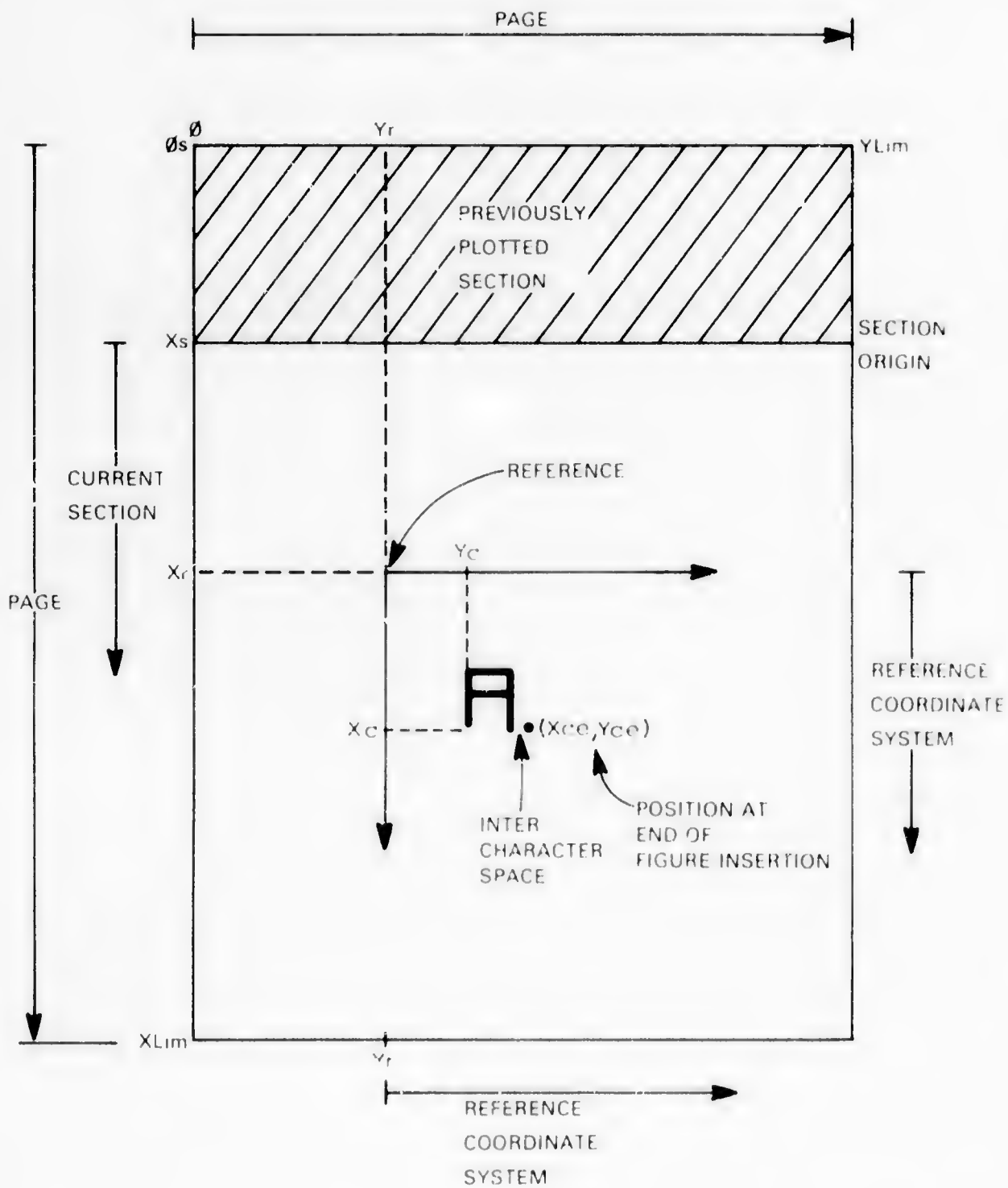


Figure 1-4 Graphics Insertion

For the LINES (VLINE, VDASH, HLINE, HDASH), the direction is implicit in the subroutine argument LENGTH which may have either a positive or negative value.

The GRAPHIC POINTER is moved in one of several ways. It may be moved:

- As the result of entering a MOVPTR request,
- As the result of entering a request to draw any graphic figure (BLOCK, BCODE, TEXT, etc.). Each graphic figure request includes parameters to reposition the GRAPHIC POINTER before actual figure insertion,
- As the result of any request that moves the REFERENCE ORIGIN (Section 1.4). Whenever the REFERENCE ORIGIN is moved, the GRAPHIC POINTER is reset to 0,0,
- As the result of action taken upon completion of the graphic figure requests (BLOCK, BCODE, etc.). BLOCK, BCODE, and TEXT leave the POINTER at the end of the character list plus the intercharacter space (1/6 of the character width), or
- As LINES (VLINE, HLINE, VDASH, HDASH) leave the pointer at the end of the line.

CHAPTER 2 GUIDE TO USING THE CREATE PROGRAM

This chapter describes how to generate graphic or plot output using the CREATE program (the interactive component of the BCP software package).

NOTE

To use the CREATE program, the user must be familiar with the use of computer terminals and the following.

- For RSX-11M BCP:
 - RSX-11M operating system, and
 - RSX-11M/M-Plus MCR Operations MANUAL, AA-L678A-TC.
- For VAX-11 BCP:
 - VAX/VMS operating system, and
 - VAX/VMS Command Language User's Guide, AA-D023C-TE.

2.1 INTRODUCTION

The CREATE user can generate GRAPHIC (plot) output (such as barcodes, block characters, etc.) by using a special English-like command language. Graphic commands are typed on the user's terminal to the CREATE program. After receiving the command, CREATE can prompt for additional information. When it has all the information it needs, it then passes the command (as a GRAPHIC REQUEST, explained in Section 1.2) onto the BCPLIB to process the command and "draw" the specified figure or figures.

2.2 ACCESSING AND USING CREATE

The terminal user accesses CREATE:

- For RSX-11M BCP -- By entering the standard MCR command RUN.

MCR>RUN CREATE
- For VAX-11 BCP -- By entering the RUN command after the system prompt (\$).

\$RUN CREATE

When becoming active via the RUN command, CREATE:

- Prompts the user for commands

CRT>

- The user can now type a command on his terminal. For example:

CRT>MOVPTR

- This sample command makes a request to move the GRAPHIC POINTER position. CREATE now prompts for additional data needed to process this command:

CRT>ENTER NEW GRAPHIC POINTER X POSITION:

CRT>ENTER NEW GRAPHIC POINTER Y POSITION:

- Now the user must respond with the new POINTER position, in whole and/or tenths of inches from the origin.

The CREATE dialogue (prompting sequence) may be bypassed when entering commands by typing a one-line command statement that includes the command parameters (data) separated by commas. For example:

CRT>MOVPTR 3.0,4.0

2.3 ENTERING CREATE COMMANDS

2.3.1 Command Formats

CREATE commands can be entered in two ways.

- The user can type the entire command and all the supporting data, or command parameters on the same line:

CRT>COMMAND Parameter 1,Parameter 2 .../OPTIONS

Example: CRT>GETFRM NAME.FRM,5.0,1.0/F

- The user can type the command name and options only. In this case, CREATE prompts the user for the additional data items required.

CRT>COMMAND/OPTIONS

VALUE PROMPT1: PARAMETER ENTRY

VALUE PROMPT2: PARAMETER ENTRY

ETC.

Example: CRT>GETFRM/F
ENTER FORM NAME (EX.BCP.FRM): BIN.FRM
ENTER NEW X REFERENCE ORIGIN: 5.0
ENTER NEW Y REFERENCE ORIGIN: 1.0

2.3.2 Parameters

A parameter is additional information that is required when a command is entered. The parameters may be included on the command line or entered later in response to a CREATE program prompt such as:

```
CRT>BCODE "ABCDE"  
      or  
CRT>BCODE  
ENTER CHARACTERS: "ABCDE"
```

2.3.3 Options

An option is additional information that is not required when a command is entered. Options are indicated by typing the character "/" followed by the option name. If the option requires a parameter, the option name is followed by "=" and the parameter value.

```
CRT>BLOCK "ABCDE"/S=1.0/D  
      or  
CRT>BLOCK /S=1.0/D  
ENTER CHARACTERS: "ABCDE"
```

Where:

S = Size (height and width of characters)

D = Double height (draw characters with height double size)

2.3.4 Data Entry

The data supplied as parameters and options are of the following four types.

- File Spec -- This is a file name and is used to define a storage area on one of the system's peripherals (disks, magtapes, etc.). For a detailed description of the file specifications and their syntax, refer to:

-- For RSX-11M BCP:

MCR Operations Manual, AA-L678A-TC, Section 2.2

-- For VAX-11 BCP:

Command Language User's Guide, AA-D023C-TE, Chapter 2

Example: CRT>NEWPLT BCP.PLT

- List -- This is a list of the alphanumeric characters required by the BCODE, BLOCK, and TEXT commands. Character lists must be preceded by the quote (") character and followed by the same character.

Example: CRT>BCODE "ABC1234"

- Decimals -- These are real numbers that include a decimal point and up to one decimal place (for example, 1.0, 1.3, 6.7, etc.).

Example: CRT>MOVREF 1.2,2.5

- Integer Numbers -- These are whole numbers that do not include a decimal point.

Example: CRT>BLOCK "ABCD"/P=3

NOTE

P is used to specify that the characters are to be inserted at the Form LABEL POINT number 3 (see Section 1.3 or the command descriptions for BLOCK and DEFPNT in Appendix A).

2.4 GUIDELINES FOR USING CREATE COMMANDS

This section familiarizes the user with the proper use of the CREATE commands needed to generate PLOT (graphic) output. The following 10 steps outline the general rules, guidelines, and order (or sequence) of command entry.

NOTE

The HELP and SHOW commands are not defined in the following outline. Both the HELP and SHOW commands can be entered at any time since they do not affect plotting.

1. To start a new plot generation session, type the command NEWPLT.

Example: CRT>NEWPLT
ENTER FILE NAME: BCP.PLT

Comments:

- All plots must start with the NEWPLT command. This command is not optional.
- NEWPLT can only follow:
 - The "RUN CREATE" command (Section 2.2), or
 - The CREATE "END" command.

To proceed from here:

Following the NEWPLT command, you can proceed with any step that follows below, except for Steps 7, 8, and 9.

2. To change the values of internal parameters, enter one or more of the following optional commands.
 - SETPAG -- To change the page limits (SIZE).
 - SETFIG -- To change:
 - The block character and barcode SIZE,
 - The barcode RATIO (2:1) or (3:1),
 - The block character double-height flag, or
 - The barcode text flag.
 - SETTAB -- To change the values of the vertical and horizontal TAB distances.

To proceed from here:

Proceed to either Step 3 or 4. If the command was SETTAB, you can proceed to Step 8.

3. To define a FORM, enter the DEFFRM and DEFPNT commands.

Example: CRT>DEFFRM
ENTER FORM FILE NAME: FIRST.FRM
ENTER FORM HEIGHT: 6.5
ENTER FORM WIDTH: 5.0

CRT>DEFPNT
ENTER POINT NUMBER: 2
ENTER X POSITION: (1-12) 1.0
ENTER POINT POSITION: 2.3

Note the following for DEFPNT:

- Enter additional (up to 12) DEFPNT commands as needed,
- Enter any combination of Step 4 (below) commands, and
- Terminate FORM with the SAVFRM command in Step 6.

Comments:

- The DEFFRM command is optional.
- This step can only be preceded by Steps 1 and 2 above.
- DEFFRM forces the BCP software to enter a FORMS definition mode of operation in which:
 - Up to 12 LABEL POINTS can be entered via the DEFPNT command.
 - All GRAPHIC REQUESTS entered at this time are considered as part of the FORM.
- The FORM graphics are entered via Step 4 commands with the exception of PUTFRM.
- Although the GETFRM command can be used while defining the FORM, its FIX /F; see GETFRM description in Appendix A) option is illegal in this mode.
- The FORM building process must be terminated by the SAVFRM command (Step 6).

To proceed from here:

Proceed with Step 4 below. Terminate FORM definition at Step 6 (SAVFRM).

4. To enter graphic figures, use one or more of the following commands.
 - BLOCK Draw block character/s.
 - BCODE Draw CODE 39 barcode.
 - TEXT Draw text characters.
 - VLINE Draw a vertical line.
 - HLINE Draw a horizontal line.
 - VDASH Draw a vertical dashed line.

- HDASH Draw a horizontal dashed line.
- GETFRM Enter a predefined form.
- PUTFRM Reuse a "fixed" form at a new position.

Comments:

When using GETFRM:

- A FORM insertion mode of operation is entered where LABEL points can be accessed.
- GETFRM is normally terminated by the ENDFRM command (Step 5).
- During FORM definition, GETFRM cannot be used with the FIX (/F; see GETFRM description in Appendix A) option.

To proceed from here:

Proceed with any of the steps that follow. However, if GETFRM or PUTFRM was entered, proceed with Step 5.

5. To terminate FORM insertion (originating from the GETFRM or PUTFRM commands), enter ENDFRM.

Example: CRT>ENDFRM

Comments:

- The GETFRM and PUTFRM commands are terminated either by ENDFRM, another GETFRM or PUTFRM command, or by the Phase 2 process (see Chapter 1) that is brought on by one of the following commands.

```
-- PROCES
-- PAGE
-- REPEAT
-- END
```
- Following ENDFRM, figure insertion using LABEL POINTS cannot be made (see the P option for the GRAPHIC commands).

To proceed from here:

- If you are in the FORM Definition mode (DEFFRM), proceed with either Step 4 or 6.
- If you are not in the FORM Definition mode, proceed with either Step 4, 7, 8, 9, or 10.

- 6 To terminate FORM definition and save the FORM, enter the SAVFRM command:

Example: CRT>SAVFRM

Comments:

SAVFRM must be used to terminate the FORM definition started by the DEFFRM command.

To proceed from here:

Proceed to either Step 2, 3, 4 or 10.

7. To terminate and process the current page section, enter PROCES.

Example: CRT>PROCES

Comments:

This is an optional command used to segment a PLOT page into smaller, more manageable sections. The current contents of the REQUEST BUFFER is processed into plottable data and written on the graphic f.le. The buffers are then cleared. The height of the graphic figures in the REQUEST BUFFER determines the section height. However, if the section height is not a multiple of 0.5 inch, it is adjusted upwards to the next 0.5 inch increment.

To proceed from here:

To proceed from here, you may go back to Step 4 or ahead to either Step 9 or 10.

8. To repeat the currently defined graphics (current contents of the REQUEST BUFFER), use the REPEAT command.

Example: CRT>REPEAT 2,3

Comments:

REPEAT makes the following assumptions:

- The intervals between the graphics being repeated is determined by the values assigned to TABH and TABV (see SETTAB command).
- TABV must be a multiple of 0.5 inch.
- REPEAT acts upon the entire contents of the REQUEST BUFFER, not just the last graphic figure entered.
- Following REPEAT by the PROCES command is unnecessary.

To proceed from here:

From here, proceed with either Step 4 or 10.

9. To terminate the current PLOT page and start a new page, enter the PAGE command.

Example: CRT>PAGE

Comments:

PAGE causes the current contents of the REQUEST BUFFERS to be processed and output. Following this, an LXY "TOP-OF-FORM" character is output to the printer. The PLOT is then reinitialized to the top of the next page. It is not necessary to enter the PROCES command to process the current contents of the REQUEST BUFFERS before PAGE.

To proceed from here:

From here, proceed with either Step 4 or 10.

10. To terminate the current PLOT session, use either of the following two commands.

- END -- Terminates PLOT and becomes available to accept a NEWPLT command to begin a new PLOT.
- EXIT -- Terminates the PLOT and the CREATE program.

Comments:

None

To proceed from here:

- END can be followed by NEWPLT to start the next PLOT.
- EXIT can be followed by the RUN CREATE command to restart the program.

2.5 CREATE POINTERS

This section supplements Section 2.4 by focusing upon actions you should take if you have special needs or if things go wrong.

2.5.1 What to Do if an Error Message Is Received Following a CREATE Command Entry

Errors can originate from two different sources. They are the:

- System (RSX-11M, VAX/VMS, FORTRAN, etc.), and
- BCP software.

Typically, errors reported by the system are fatal and the CREATE program aborts and the current PLOT is lost.

The BCP software detects nonfatal system errors, CREATE command errors, and miscellaneous PLOT errors such as page limit violations. These errors are reported to the user on the terminal (see Appendix C) in English messages. In this case, the last CREATE command that caused the error situation is aborted with no change in either the state of the program or the graphic output. You may continue from here with further command entry.

2.5.2 How to Generate a Multipage Plot Solution:

- Use the PAGE command (rather than PROCES) for the last section of each page.
- However, if the REPEAT command is used, there is no need to use the PAGE command. REPEAT automatically calls the PAGE command at the appropriate time.

2.5.3 If the REQUEST BUFFER is Filled

The REQUEST BUFFER can hold up to 132 graphic commands (BLOCK characters, BARCODES, etc.). The actual number varies with the amount of information that is stored by the individual requests. Some requests require more space than others. The GETFRM command, for example, can bring in a large, complex form requiring a lot of space.

For the typical PLOT, the buffers should be adequate to hold the entire PLOT PAGE.

If your PLOT is particularly dense or you have to use a number of FORMS, then you should plan a strategy after considering the following.

- Can you divide your PLOT into more than one PAGE? The current PLOT PAGE is terminated and the paper is advanced to the top of the next page whenever the PAGE command is entered. Thus the REQUEST BUFFERS are cleared and available to store the next PAGE requests.
- Can you divide your PLOT PAGE into smaller, more manageable SECTIONS? The CREATE command PROCES terminates the page SECTION and causes the currently stored requests in the buffers to be processed into plottable data for output. The buffer is then cleared, and becomes available to accept requests for the next SECTION.

- If the same form is reused on the PLOT PAGE (GETFRM Command), consider "FIXing" the form into memory (GETFRM's FIX option) and use PUTFRM to place the form at the different page positions needed. In this way, the FORMS graphic contents are stored only once instead of once for each position where the FORM is needed.
- Can you make use of the REPEAT command? REPEAT is designed to repeat the current contents of the REQUEST BUFFERS up to approximately 32,000 times in the vertical direction. No additional space is required in the buffers. PAGES are recognized and the line printer page is automatically advanced by the TOP-OF-FORM command at the end of each page.

2.5.4 To Erase Previously Entered Commands

The ERASE command can be used to clear the entire REQUEST BUFFER, but not an individual command. For this reason, a strategy of generating PLOTS in a series of smaller SECTIONS should be considered.

2.5.5 To Remove a Fixed Form

The ERASE command is the only way to do this. ERASE has an "erase fixed form only" option (ERASE FIX; see Appendix A descriptions).

2.5.6 To Get Plot Positioning and Other Useful Information

The SHOW command types the following information on the user's terminal (refer to Section A.23 for format).

- REFERENCE position (MOVREF)
- GRAPHICS POINTER position
- PAGE parameters (SETPAG)
- FIGURE parameters (SETFIG)
- FORM parameters (GETFRM)

2.5.7 To Request a Summary of CREATE Commands

Type HELP

APPENDIX A CREATE LISTING

Appendix A is an alphabetical listing of the CREATE commands including format, prompts, options, examples, and descriptions.

A.1 BCODE

Draw a CODE 39 barcode from a specified list of characters.

Format:

BCODE [List]	
<u>Command Options</u>	<u>Defaults</u>
/S = Size	/S = 1.0 or value assigned by SETFIG command
/P = Number	
/I = x,y	Current GRAPHIC POINTER position
/A = x,y	Current GRAPHIC Pointer position

Prompts:

ENTER CHARACTERS: List

Command Parameters:

List -- List of characters are represented by the BARCODE. The list of characters must be enclosed in quotation marks. If List is typed with the BCODE command, the prompt above is bypassed.

Legal characters are: 0-9
 A-Z
 \$.+ - / %

Description:

The BCODE command requests to draw (or plot) a list of characters in CODE 39 barcode format. The barcode height is determined by the internal parameter size or by the /S command option. The initial value of size is 1.0 inch, however, this value can be changed by the use of the SETFIG command.

The barcode is drawn with a ratio (wide line widths to thin line widths and wide spaces to thin spaces) of either 2:1 or 3:1. These ratios are dependent on the value of the internal parameter RATIO. RATIO has an initial value of 2, but use of the SETFIG command can change the value to 3.

Barcode text can be printed along with the barcode figure, but the barcode height is shortened by 0.2 inch. The SETFIG command can be used to delete the barcode text (OMIT parameter) when plotting barcodes.

The barcode is inserted at the current GRAPHIC POINTER position, starting at the lower left corner of the barcode figure itself. Upon completion of the BCODE command, the GRAPHIC POINTER is left at the lower right corner of the figure.

Options:

- /A=x,y The GRAPHIC POINTER is moved to a new location determined by x and y, that are defined relative to the current REFERENCE ORIGIN before the insertion of the block characters. Both x and y are decimal numbers expressing the new position in whole and/or tenth of an inch units.
- /I=x,y The GRAPHIC POINTER is moved a distance, defined by x and y, to a new position before the insertion of the block characters. Both x and y are decimal numbers expressing distances in whole and/or tenth inch units.
- /P=n This option assumes that CREATE is in a FORMS insertion mode initiated by the GETFRM (or PUTFRM) command and terminated by the ENDFRM, PROCESS, PAGE, or REPEAT command. A LABEL POINT "n" on the current form is accessed and the GRAPHIC POINTER is moved to the position defined by the point. The "n" is an integer having a value in the range from 1 to 12.
- /S=Size Draw the barcode of size "Size". Size is a decimal number expressing the barcode height in whole and/or tenth of an inch units.

Examples:

1. CRT>BCODE/S=5.0/I=6.0,1.0
ENTER CHARACTERS: "ABCD"

Generates a CODE 39 type barcode for the characters ABCD. The BARCODE is drawn 5.0 inches high at a position 6.0 inches down the page (x) and 1.0 inch across the page (y) from the current GRAPHICS POINTER position.

2. CRT>BCODE "A123Z"/P=5

Generate a CODE 39 type barcode for the characters A123Z at the current form's 5th LABEL POINT. (See /P description above.)

A.2 BLOCK

Draw one or more block characters.

Format:

BLOCK [character list]

<u>Command Options</u>	<u>Defaults</u>
/S = Size	/S=1.0 or value assigned by SETFIG Command
/P = Number	
/I = x,y	Current GRAPHIC POINTER position
/A = x,y	Current GRAPHIC POINTER position
/D = Double Height	Height = Width (Size)

Prompts:

ENTER CHARACTERS: character list

Command Parameters:

character list -- List of characters are displayed as BLOCK characters. Character List must be enclosed in quotation marks. The above prompt is bypassed if the Character List is typed on the same line as the BLOCK command.

Legal Characters are: 0-9
A-Z
%.'*-/,,:;=?%

Description:

The BLOCK command makes a request to draw (or plot) a list of characters in block format. Block formatted characters are drawn such that the character height and the width are the same unless the double height option (/D) is specified. For double height, the character height is twice the width. The character width includes intercharacter spacing on the character's right side which is one sixth of the total width.

Unless the size option (/S) is specified, the character height and width is determined by the internal parameter by the same name (size). Initially, size has the value of 1.0 inch, however, its value may be changed by the SETFIG command.

The characters are inserted at the current GRAPHIC POINTER position starting at the lower left corner of the first character.

Upon completion of the BLOCK command, the GRAPHIC POINTER is left at the last character's lower right corner (including the intercharacter space).

Options:

- /Size=Size Draw a BLOCK character(s) of size "Size". The default value for Size is initially 1.0 (see SETFIG command).
- /D Draw character(s) double height. Double height can be used in conjunction with the other BLOCK options.
- /A=x,y The GRAPHIC POINTER is moved to a new location determined by x and y, which are defined relative to the current REFERENCE ORIGIN before the insertion of the block characters. Both x and y are decimal numbers expressing the new position in whole and/or tenth inch units.
- /I=x,y The GRAPHIC POINTER is moved a distance defined by x and y, to a new position before the insertion of the block characters. Both x and y are decimal numbers expressing distances in whole and/or tenth inch units.
- /P=n This option assumes that CREATE is in a forms insertion mode initiated by the GETFRM (or PUTFRM) command and is terminated by the ENDFRM, PROCESS, PAGE, or REPEAT commands. A LABEL POINT "n" on the current form is accessed and the GRAPHIC POINTER is moved to the position defined by the point. The "n" is an integer having a value in the range from 1 to 12.

Example:

1. CRT>BLOCK/S=1.5/D/I=6.0,1.0
ENTER CHARACTERS: "BCP"

The block characters BCP are drawn at a position 6.0 inches down the page (x) and 1.0 inch across the page (y) from the current GRAPHICS POINTER position. The block characters are a total of 3.0 inches high since the size option was set for 1.5 inches and the double-height option was specified.

A.3 DEFFRM

Initiate FORM GENERATION (or definition).

Format:

```
DEFFRM [form name,height,width]
```

Prompts:

ENTER FORM FILE NAME: form name

ENTER FORM HEIGHT: height

ENTER FORM WIDTH: width

Command Parameters:

form name -- Enter a form name of 1-27 characters. The form name is the file specified for the disk file where the form is stored. (Refer to RSX-11M/M-Plus MCR Operations Manual, Section 2.2. or VAX/VMS Command Language User's Guide, Chapter 2.)*

height -- Enter the desired height of the form being created. The height defines the lower boundary of the form when inserting the form graphics.

width -- Enter the desired width of the form being created. The width defines the right boundary of the form when inserting the form graphics.

Description:

The DEFFRM command initiates a forms definition mode of operation where the user may enter commands describing the form in terms of lines, barcodes, text, and block characters. These commands or requests are then saved for later use on a disk file. The form may also include LABEL POINTS defined by the DEFPNT command. When the form is completed, the user enters the SAVFRM command to save the form and its contents on a special disk file created for this purpose.

Forms can not be defined after normal plotting is started.

Options:

None

* The user should supply the .FRM extension to the form name. If a name is given without an extension, the system supplies the extension .DAT.

Examples:

1. CRT>DEFFRM
ENTER FORM FILE NAME: BCP.FRM
ENTER FORM HEIGHT: 5
ENTER FORM WIDTH: 5

Defines a form named BCP.FRM with a height of 5.0 inches and a width of 5.0 inches. GRAPHICS COMMANDS (such as BLOCK, BCODE, etc.) can be entered and LABEL POINTS can be defined (see DEFPNT) at this time.

2. CRT>DEFFRM FORM.FRM,3.0,4.5

Defines a form named FORM.FRM with a height of 3.0 inches and a width of 4.5 inches.

A.4 DEFPNT

Format:

DEFPNT [Point Number,dx,dy]

Prompts:

ENTER POINT NUMBER (1-12): Point Number

ENTER X POSITION: dx

ENTER Y POSITION: dy

Command Parameter:

Point Number -- Enter the point number (1-12) being defined.

dx -- Enter the x position (relative to the form base) of the LABEL POINT.

dy -- Enter the y position (relative to the form base) of the LABEL POINT.

Description:

DEFPNT defines a FORM LABEL POINT. LABEL POINTS are a set of up to 12 predefined locations on the form that can be called out by number (1-12) to place the additional graphics at form insertion time. DEFPNT may only be entered while in the FORMS DEFINITION mode (that is initiated by the GETFRM command and terminated by the SAVFRM command).

Options:

None

Example:

```
1. CRT>DEFPNT
   ENTER POINT NUMBER (1-12): 2
   ENTER X POSITION: 1.5
   ENTER Y POSITION: 3.2
```

Defines LABEL POINT number 2 at a position 1.5 inches down the form (x) and 3.2 inches across the form (y) from the form's base position (upper left corner).

A.5 END

Terminate current plot

Format:

END

Prompts:

None

Command Parameters:

None

Description:

The END command closes a plot file opened by the NEWPLT command. After an END command, CREATE can accept a NEWPLT request to begin a new plot or an EXIT request to terminate the CREATE program.

Options:

None

Example:

CRT>END

The END command terminates the current plot. At this point, the user can exit from CREATE or initialize a new plot.

A.6 ENDFRM

Terminate form insertion.

Format:

ENDFRM

Prompts:

None

Command Parameters:

None

Description:

The ENDFRM command terminates form insertion mode initiated by either the GETFRM or PUTFRM command. Once ENDFRM is entered, the FORMS LABEL POINT can no longer be accessed.

Options:

None

Example:

CRT>ENDFRM

The ENDFRM command terminates form insertion. This command is valid only after the GETFRM or PUTFRM commands.

A.7 ERASE

ERASE commands back to last PROCES or NEWPLT command.

Format:

ERASE BUF

Prompts:

None

Command Parameters:

BUF -- Specifies what buffer to erase. BUF may be one of the following three-character words.

- FIX -- Erase a form fixed in memory.
- REQ -- Erase the contents of the REQUEST BUFFER.
- ALL -- Erase both the contents of the REQUEST BUFFER and a form fixed in memory.

Description:

The ERASE command can erase a form fixed in memory, the current REQUEST BUFFER, or both. The command structure, parameters, and descriptions of each are listed below.

- ERASE FIX -- Erases a form fixed in memory (see the GETFERM/F command). ERASE FIX must be used before another GETFERM command can be used.
- ERASE REQ -- Erases the current REQUEST BUFFER to the last PROCES, PAGE, or REPEAT command or back to NEWPLT.
- ERASE ALL -- Erases both the current REQUEST BUFFER and the form currently fixed in memory.

Options:

None

Examples:

1. CRT>ERASE

The ERASE command was entered with no parameters. The default, in this case, is to ERASE ALL; for example, both the contents of the REQUEST BUFFER and any form fixed in memory is erased.

2. CRT>ERASE FIX

The form currently fixed in memory is erased.

3. CRT>ERASE REQ

The contents of the current REQUEST BUFFER (back to the last PROCES, PAGE, or REPEAT command or back to NEWPLT) is erased.

A.8 EXIT
Exit program.

Format:

EXIT

Prompts:

None

Command Parameters:

None

Description:

The EXIT command terminates the CREATE program.

Options:

None

Example:

CRT>EXIT

This causes the user to EXIT from the CREATE program.

A.9 GETFRM

Retrieve a FORM for use.

Format:

GETFRM [filename,xref,yref]	
<u>Command Option</u>	<u>Defaults</u>
/F	NO FIX

Prompts:

ENTER FORM NAME (EX.BCP.FRM): filename

ENTER NEW X REFERENCE POINT: xref

ENTER NEW Y REFERENCE POINT: yref

Command Parameters:

filename -- Filename of the FORM being retrieved.

xref -- New x REFERENCE point where retrieved form is placed.

yref -- New y REFERENCE point where retrieved form is placed.

Description:

GETFRM reads a stored FORM from a disk file and inserts it in the REQUEST buffer in the same manner as a normal GRAPHIC REQUEST such as BCODE. In addition, GETFRM initiates a FORMS insertion mode of operation that permits accessing of LABEL POINTS by the graphic commands via the /P option (see BCODE, BLOCK, TEXT, etc. command descriptions).

The FORMS insertion mode is terminated by one of the following commands.

- Another GETFRM command
- ENDFRM
- PROCES
- PAGE
- REPEAT
- END

To save or fix the FORM contents in memory for shared use, the option /F can be used. A fixed form can be reused for any number of positions on the page via the PUTFORM command from a single copy of its graphic contents stored in this way.

Whenever a FORM is fixed into memory, another GETFORM cannot be entered until the FORM is removed. A fixed form is removed only by the ERASE command using the FIX parameter.

Options:

/F -- Fix or store the retrieved form in memory so that it can be used repeatedly.

Examples:

1. CRT>GETFORM/F
ENTER FORM NAME (EX.BCP.FRM): FORM.FRM
ENTER NEW X REFERENCE POINT: 1.0
ENTER NEW Y REFERENCE POINT: 0.0

The form FORM.FRM is retrieved and fixed in memory. It also is at a point 1.0 inch down the page (x) and 0.0 inch across the page (y). The PUTFORM command can now be used to make any number of additional insertions of the form on the plot output. Graphics can now be placed at the predefined LABEL POINTS using graphics commands with the /P options.

2. CRT>GETFORM MYFORM.FRM,3,2/F

The form MYFORM.FRM is fixed in memory. MYFORM.FRM is positioned on the page at a point 3.0 inches down and 2.0 inches across the page. The prompt message is bypassed.

3. CRT>GETFORM ABC.FRM,1,1

The form ABC.FRM is positioned at the new REFERENCE position 1.0 inch down the page and 1.0 inch across the page. The form is not fixed in memory. The prompt message was bypassed since the desired form name was entered with the GETFORM command.

A.10 HDASH

Draw a horizontal dashed line.

Format:

HDASH [length]	
<u>Command Options</u>	<u>Defaults</u>
/P=Number	
/I=x,y	Current GRAPHIC POINTER position
/A=x,y	Current GRAPHIC POINTER position
/T=Thickness	Normal dashed line thickness

Prompts:

ENTER LENGTH: length

Command Parameters:

length -- Desired length of the line in inches. The prompt does not accept a carriage return since there is no default length for a line.

Description:

The HDASH command makes a request to draw (or plot) a horizontal dashed line. The length of the line is specified by the user.

The lines are inserted at the current GRAPHIC POINTER position.

Upon completion of the HDASH command, the GRAPHIC POINTER is left at the end of the line or the lower right corner of a thick line.

Options:

/P=n -- Place the line at the current FORMS LABEL POINT number "n" where "n" ranges between 1 and 12. This option is legal only during the FORMS insertion mode (see GETFRM).

/I=x,y -- Place the line at an incremental distance x,y from the current GRAPHIC POINTER position.

/A=x,y -- Place the line at a distance x,y from the current REFERENCE and update the GRAPHIC POINTER position to this point.

/T=Thickness -- Specify the desired thickness of the line. Thickness can range from 0.1 inch to the limits of the page. If /T is not called, the line defaults to a thickness of one dot on the printer/plotter (a normal line).

Examples:

1. CRT>HDASH/P=4/T=.2
ENTER LENGTH: 4

A dashed line with a length of 4.0 inches and a thickness of 0.2 inch is entered at the current form's 4th LABEL POINT. (See the /P description above for restrictions.)

2. CRT>HDASH 3/I=1,2.5

A dashed line with a length of 3.0 inches is entered at a position 1.0 inch down (x) and 2.5 inches across (y) from the current GRAPHIC POINTER position.

3. CRT>HDASH 2.5/A=3,2/T=.4

A dashed line with a length of 2.5 inches and a thickness of 0.4 inch is entered at a position 3.0 inches down (x) and 2.0 inches across (y) from the current GRAPHIC REFERENCE position.

A.11 HELP

Type help information.

Format:

HELP

Prompts:

None

Command Parameters:

None

Description:

The HELP command displays the file HELPCR.DAT at the user's terminal. This is an abbreviated help file showing the command string format and the options used with each command.

Options:

None

Example:

CRT>HELP

A HELP file, giving the command format and the listing options for each command, is printed at the screen or terminal.

A.12 HLINE

Draw a horizontal line.

Format:

HLINE [Length]	
<u>Command Qualifier</u>	<u>Defaults</u>
/P=N	
/I=x,y	Current GRAPHIC POINTER position
/A=x,y	Current GRAPHIC POINTER position
/T=Thickness	Normal line thickness

Prompts:

ENTER LENGTH: Length

Command Parameters:

Length -- Desired length of the line in inches. The prompt does not accept a carriage return since there is no default length for a line.

Description:

The HLINE command makes a request to draw (or plot) a horizontal line. The length of the line is specified by the user.

The lines are inserted at the current GRAPHIC POINTER position.

Upon completion of the HLINE command, the GRAPHIC POINTER is left at the end of the line or at the lower right corner of a thick line.

Options:

/P=n -- Place the line at the current FORMS LABEL POINT number "n" where "n" ranges between 1 and 12. This option is legal only during the FORMS INSERTION MODE (see GETFRM command).

/I=x,y -- Place the line at an incremental distance x,y from the current GRAPHIC POINTER position.

/A=x,y --- Place the line at a distance x,y from the current REFERENCE and update the GRAPHIC POINTER position to this point.

/T=Thickness -- Specify the desired thickness of the line. Thickness can range from 0.1 inch to the limits of the page. If /T is not called, the line defaults to a thickness of one dot on the printer/plotter (a normal line).

Examples:

1. CRT>HLINE/P=4/T=.2
ENTER LENGTH: 4

A line with a length of 4.0 inches and a thickness of 0.2 inch is entered at the current form's 4th LABEL POINT. (See the /P description above for restrictions.)

2. CRT>HLINE 3/I=1,2.5

A line with a length of 3.0 inches is entered at a position 1.0 inch down (x) and 2.5 inches across (y) from the current GRAPHIC POINTER position.

3. CRT>HLINE 2.5/A=3,2/T=.4

A line with a length of 2.5 inches and a thickness of 0.4 inch is entered at a position 3.0 inches down (x) and 2.0 inches across (y) from the current GRAPHIC REFERENCE position.

A.13 HTAB

Move the REFERENCE ORIGIN horizontally by a specified number of TAB increments.

Format:

HTAB [hnun]

Prompts:

ENTER NUMBER OF HORIZONTAL TABS: hnun

Command Parameters:

hnun -- Number of horizontal tabs to move the REFERENCE ORIGIN.

Description:

The HTAB command moves the REFERENCE ORIGIN by a multiple of the horizontal tab distance (TABH; see SETTAB command) across the page. The distance between tabs (set by the SETTAB command) is checked to see that the limits of the page are not violated before moving the reference.

Options:

None

Examples:

1. CRT>HTAB
ENTER NUMBER OF TABS: 5
2. CRT>HTAB 5

A.14 MOVPTR
Move GRAPHIC POINTER.

Format:

MOVPTR [x,y]

Prompts:

ENTER NEW GRAPHIC POINTER X POSITION: x
ENTER NEW GRAPHIC POINTER Y POSITION: y

Command Parameters:

x -- New GRAPHIC POINTER X position is at coordinate x.
y -- New GRAPHIC POINTER Y position is at coordinate y.

Description:

The MOVPTR request moves the GRAPHIC POINTER position x inches down the page and y inches across the page from the current REFERENCE origin.

Options:

None

Examples:

1. CRT>MOVPTR
ENTER NEW GRAPHIC POINTER X POSITION: 2
ENTER NEW GRAPHIC POINTER Y POSITION: 3.5

The GRAPHIC POINTER position is moved to the point 2.0 inches down the page (x) and 3.5 inches across the page (y) from the REFERENCE ORIGIN.

2. CRT>MOVPTR 1.7,1.3

The GRAPHIC POINTER position is moved to the point 1.7 inches down the page (x) and 1.3 inches across the page (y) from the REFERENCE ORIGIN.

A.15 MOVREF

Reposition REFERENCE ORIGIN.

Format:

MOVREF [x,y]

Prompts:

ENTER NEW REFERENCE X POSITION: x
ENTER NEW REFERENCE Y POSITION: y

Command Parameters:

x -- New REFERENCE ORIGIN X coordinate.
y -- New REFERENCE ORIGIN Y coordinate.

Description:

The MOVREF request moves the REFERENCE ORIGIN to a point x inches down the page and y inches across the page from the PAGE origin (absolute 0,0).

Options:

None

Examples:

1. CRT>MOVREF
ENTER NEW REFERENCE X POSITION: 2.5
ENTER NEW REFERENCE Y POSITION: 1

The REFERENCE ORIGIN is repositioned at a point 2.5 inches down (x) and 1.0 inch across (y) the page from point 0,0.

2. CRT>MOVREF 1,1

The REFERENCE ORIGIN is repositioned at a point 1.0 inch down (x) and 1.0 inch across (y) the page from point 0,0.

A.16 NEWPLT

Prepare for a new plot output.

Format:

NEWPLT [File name]

Prompts:

ENTER FILENAME: File name

Command Parameters:

File name -- File name is any legal PDP-11 or VAX/VMS file specification (such as, DR0:BCP.PLT) for the graphics file. This can be any legal file name portion of the specification (for example, BCP).

Description:

The NEWPLT request opens a new file for plot output. The user can specify a name or, by answering the prompt with a carriage return, can choose the default name (BCP.PLT). If the user chooses to specify a name, the extension .PLT should be used. If no extension is provided, the system default of .DAT is provided.

Options:

None

Examples:

1. CRT>NEWPLT BLOCK.PLT

A file named BLOCK.PLT is opened.

2. CRT>NEWPLT

ENTER FILENAME: GRAPH.PLT

A file named GRAPH.PLT is opened.

A.17 PAGE

Terminate the current PLOT page and start a new page.

Format:

PAGE

Prompts:

None

Command Parameters:

None

Description:

PAGE causes the current contents of the REQUEST BUFFERS to be processed and output. Following this, an LXY "TOP-OF-FORM" character is output to the printer. The PLOT is then reinitialized to the top of the next page. It is not necessary to enter the PROCES command to process the current contents of the REQUEST BUFFERS before PAGE.

Example:

CRT>PAGE

A.18 PROCES

Output current page section.

Format:

·PROCES

Prompts:

None

Command Parameters:

None

Description:

The PROCES command is used to force the Phase 2 processing of the stored REQUESTS in the REQUESTS BUFFERS into plottable data. The plottable data is stored on a special file, called the graphics file, for later transfer to the LXY printer to plot the graphics. The PROCES command defines the SECTION that is the area of the page covered by the figures defined by the stored REQUESTS. The SECTION height is determined by the height of the resultant height of the graphic or plotted figures defined by the stored REQUESTS. The SECTION height is adjusted upwards to multiple of 0.5 inch.

Example:

CRT>PROCES

A.19 PUTFRM

Reuse a "Fixed" form at a new position.

Format:

PUTFRM [xoff, yoff]

Prompts:

ENTER REFERENCE OFFSET IN X: xoff

ENTER REFERENCE OFFSET IN Y: yoff

Command Parameters:

xoff -- New x REFERENCE offset point where retrieved form is placed.

yoff -- New y REFERENCE offset point where retrieved form is placed.

Description:

PUTFRM places a fixed form at any number of places on the output PAGE. PUTFRM cannot be used unless a form is fixed in memory (see GETFRM).

The FORMS insertion mode is terminated by one of the following commands.

- ENDFRM
- PROCES
- PAGE
- REPEAT
- END

Whenever a FORM is fixed into memory, it must be removed by the ERASE command using the FIX parameter before another GETFRM command can be used.

Options:

None

Examples:

1. CRT>PUTFRM
ENTER REFERENCE OFFSET IN X: 4.0
ENTER REFERENCE OFFSET IN Y: 1.0

The FORM currently fixed in memory is placed at a point 4.0 inches down the page (x) and the 1.0 inch across the page (y). The PUTFRM can be used to make any number of additional insertions of the FORM on the plot output.

2. CRT>PUTFRM 6,7

The FORM currently fixed in memory is inserted at a point 6.0 inches down the page (x) and 7.0 inches across the page (y). The prompt message is bypassed.

A.20 REPEAT

Output the graphics requested up until this point. Repeat them as specified.

Format:

REPEAT [hrep,vrep]

Prompts:

ENTER NUMBER OF HORIZONTAL REPEATS: hrep
ENTER NUMBER OF VERTICAL REPEATS: vrep

Command Parameters:

hrep -- Number of horizontal repeats desired.

NOTE

TABH (see SETTAB) must be set before REPEAT is called.

vrep -- Number of vertical repeats desired.

NOTE

TABV (see SETTAB) must be set before REPEAT is called.

Description:

The GRAPHIC REQUESTS currently in the REQUEST BUFFER are repeated hrep times horizontally and vrep times vertically. REPEAT calls on the PROCES and PAGE facilities as needed to process the requests by SECTION and to advance the PLOT PAGE.

The requests are repeated at intervals that are defined by the system TAB parameters (horizontal and vertical) that can be set up by the SETTAB command.

NOTE

It is important for the user to set up appropriate values for TABH and TABV by entering the SETTAB command. The value for TABV must be a multiple of 0.5 inch when used with REPEAT.

Options:

None

Example:

CRT>REPEAT 3,2

Repeats graphics in current REQUEST BUFFER 3 (four actual entries) times across the page and 2 (three actual entries) times down the page (actual total of 12 entries) using the spacing set up by the SETTAB command.

A.21 SAVFRM

Terminate form generation.

Format:

SAVFRM

Prompts:

None

Command Parameters:

None

Description:

The SAVFRM request terminates FORM DEFINITION and writes the contents of the REQUEST BUFFER onto the FORM file for later use by the GETFRM command. This command is valid only after the DEFFRM command.

Options:

None

Example:

CRT>SAVFRM

Terminate the FORM being generated by DEFFRM.

A.22 SETFIG

Set up default figure parameters.

Format:

SETFIG [size,ratio,dbht,omit]

Prompts:

ENTER CHARACTER SIZE: size
ENTER BARCODE RATIO (2 OR 3): ratio
DO YOU WANT DOUBLE HEIGHT FIGURES (Y/N): dbht
DO YOU WANT TO OMIT BARCODE TEXT (Y/N): omit

Command Parameters:

size -- Set the character size.

ratio -- Set a barcode ratio of 2 (2:1) or 3 (3:1).

dbht* -- Set double-height figures. Default is a regular height, where both character height and width are equal to the value of "size".

omit* -- Omit barcode text beneath barcode.

Description:

The SETFIG command provides the user with a means to change the system figure parameters used for block character and barcode generation. When the BLOCK and BCODE commands are entered, and the /S option is not used, the figure size used is the system figure size. When the BLOCK command is entered, and the /D option is not used, the value of the dbht is used by the software. RATIO and OMIT are used to determine the format of barcode.

Options:

None

*If anything other than "Y" is entered, it is taken as a no answer.

Examples:

1. CRT>SETFIG
ENTER CHARACTER SIZE: 1.5
ENTER BARCODE RATIO (2 OR 3): 3
DO YOU WANT DOUBLE HEIGHT FIGURES (Y/N): Y
DO YOU WANT TO OMIT BARCODE TEXT (Y/N): Y
2. CRT>SETFIG 3,2,N,N

Where:

- 3 -- Sets the default character size to 3.0 inch.
- 2 -- Sets the default barcode ratio to 2:1.
- N -- (NO) Figure is regular height instead of double height.
- N -- (NO) Text is included with all barcodes.

A.23 SETPAG

Format:

SETPAG [length,width,evfu]

Prompts:

ENTER PAGE LENGTH: length
ENTER PAGE WIDTH: width
DO YOU WANT TO BYPASS THE EVFU (Y/N): evfu

Command Parameters:

length -- Page length for limit checks and PAGE command use. When EVFU UNIT is used, LENGTH cannot be greater than 22.0 inches, and must be expressed to the nearest 0.5 inch increment.

width -- Page width for limit checks. Resolution 0.1 inch (13.2 limit).

evfu -- Logical flag. If evfu is Y (TRUE), bypass is set up (if anything other than "Y" is typed, the answer is set to NO).

Description:

The LXY PRINTER/PLOTTER has a facility called EVFU that is set up by the BCP software. EVFU determines how far the paper is to advance when a "TOP-OF-FORM" is received, either mechanically by the "TOP-OF-FORM" switch or by BCP software. The BCP software sets up the unit according to the page length (LENGTH), so that when the PAGE command is entered, the paper advances up to the distance set for LENGTH (depending on how far the PLOT is down the page).

Options:

None

Examples:

1. CRT>SETPAG 7.5,10.,N

The page length is set to 7.5 inches and the width is set to 10.0 inches. EVFU is not bypassed.

2. CRT>SETPAG
ENTER PAGE LENGTH: 24
ENTER PAGE WIDTH: 13.2
DO YOU WANT TO BYPASS THE EVFU (Y/N): Y

The page length is set to 24.0 inches, therefore, the EVFU must be bypassed. The width is set to 13.2 inches.

A.24 SETTAB

Set up tab distance parameters.

Format:

SETTAB [tabh,tabv]

Prompts:

ENTER HORIZONTAL TAB DISTANCE: tabh

ENTER VERTICAL TAB DISTANCE: tabv

Command Parameters:

tabh -- Set the distance for horizontal tabs.

tabv -- Set the distance for vertical tabs. Must be multiple of 0.5 inch when used with REPEAT.

Description:

The SETTAB request sets the spacing intervals between horizontal and vertical tabs. The length between tabs cannot go beyond the current page limits. These TAB parameters are later used to:

- Move the REFERENCE ORIGIN by these intervals (TABV, TABH) whenever the Tab command is used, and
- Set up REPEAT intervals both horizontally (tabh) and vertically (tabv), for use by the REPEAT command.

Options:

None

Examples:

1. CRT>SETTAB
ENTER HORIZONTAL TAB DISTANCE: 4.5
ENTER VERTICAL TAB DISTANCE: 3.0

Horizontal tabs are set across the page every 4.5 inches. Vertical tabs are set down the page every 3.0 inches. (NOTE: This command must be called before calling the REPEAT command.)

2. CRT>SETTAB 2,2

Horizontal tabs are set across the page at 2.0 inch intervals. Vertical tabs are set down the page, also at 2.0 inch intervals.

A.25 SHOW

Type current values of system parameters.

Format:

SHOW [Type]

Prompts:

None

Command Parameters:

Type:

The parameter types are as follows.

POS -- Position parameters
PAG -- Page parameters
FIG -- Figure parameters
FOR -- Form parameters
ALL -- All parameters

NOTE

If no Type is entered, SHOW defaults to SHOW ALL.

Where:

POS returns:

REFERENCE ORIGIN...X,Y
CURRENT POSITION...X,Y
CURRENT SECTION...X

PAG returns:

PAGE LENGTH
PAGE WIDTH
TABV DISTANCE
TABH DISTANCE

FIG returns:

CHARACTER SIZE
DOUBLE HEIGHT (Y/N)
RATIO (2,3)
OMIT BARCODE TEXT (Y/N)

FOR returns:

FORM WIDTH
FORM LENGTH
12 FORM LABEL POINT POSITIONS (X,Y VALUES)

ALL returns:

ALL THE ABOVE

Description:

The SHOW command displays current and default parameters. SHOW has five parameters listed and described below.

SHOW POS	--	Position parameters
SHOW PAG	--	Page parameters
SHOW FIG	--	Figure parameters
SHOW FOR	--	Form parameters
SHOW ALL	--	All of the above parameters (in the order listed)

Options:

None

Examples:

1. CRT>SHOW

SHOW defaults to SHOW ALL. A list of all the above parameters is shown.

2. CRT>SHOW POS

The position parameters are displayed.

A.26 TEXT

Print 0.1 inch block characters.

Format:

TEXT [List of characters]	
<u>Command Options</u>	<u>Defaults</u>
/P=N	
/I=X,Y	Current GRAPHIC POINTER position
/A=DX,DY	Current GRAPHIC POINTER position

Prompts:

ENTER TEXT: List of characters

Command Parameters:

List of characters -- Character list enclosed in double quotation marks that have a printed height of 0.1 inch.

Description:

The TEXT command allows the user to insert a list of standard text characters 0.1 inch high on the PLOT PAGE.

Options:

- /P=n Place the character list at the current FORMS LABEL POINT "n" where "n" ranges between 1 and 12. This option is legal only during the FORMS INSERTION mode.
- /I=x,y Place the character list at an incremental distance X,y from the current GRAPHIC POINTER position.
- /A=dx,dy Place the character list at a distance x,y from the current REFERENCE and update the GRAPHIC POINTER position to this point.

Examples:

1. CRT>TEXT
ENTER TEXT: "BARCODE SOFTWARE"

The Character List "BARCODE SOFTWARE" is printed at the current POINTER position.

2. CRT>TEXT "BCP"/P=4

The Character List "BCP" is printed at the current form's 4th LABEL POINT.

3. CRT>TEXT "TEST 1234"/I=4,2

The Character List "TEST 1234" is printed at a position 4.0 inches down the page (x) and 2.0 inches across the page (y) from the current GRAPHIC POINTER position.

A.27 VDASH

Draw a vertical dashed line.

Format:

VDASH [length]	
<u>Command Options</u>	<u>Defaults</u>
/P=Number	
/I=x,y	Current GRAPHIC POINTER position
/A=x,y	Current GRAPHIC POINTER position
/T=Thickness	Thin dashed line

Prompts:

ENTER LENGTH: length

Command Parameters:

length -- Desired length of the line in inches. the prompt does not accept a carriage return since there is no default length for a line.

Description:

The VDASH command makes a request to draw (or plot) a vertical dashed line. The length of the line is specified by the user.

The lines are inserted at the current GRAPHIC POINTER position.

Upon completion of the VDASH command, the GRAPHIC POINTER is left at the end of the line or the lower right corner of a thick line.

Options:

/P=n Place the line at the current form's LABEL POINT number "n" where "n" ranges between 1 and 12. This option is legal only during the FORMS INSERTION mode (see GETFRM command).

/I=x,y Place the line at an incremental distance x,y from the current GRAPHIC POINTER position.

/A=x,y Place the line at a distance x,y from the current REFERENCE and update the GRAPHIC POINTER position to this point.

/T=Thickness Specify the desired thickness of the line. Thickness can range from 0.1 inch to the limits of the page. If /T is not called, the line defaults to a thickness of one dot on the printer/plotter (a normal line).

Examples:

1. CRT>VDASH/P=4/T=.2
ENTER LENGTH: 4

A dashed line with a length of 4.0 inches and a thickness of 0.2 inch is entered at the current form's 4th LABEL POINT. (See the /P description above for restrictions.)

2. CRT>VDASH 3/I=1,2.5

A dashed line with a length of 3.0 inches is entered at a position 1.0 inch down (x) and 2.5 inches across (y) from the current GRAPHIC POINTER position.

3. CRT>VDASH 2.5/A=3,2/T=.4

A dashed line with a length of 2.5 inches and a thickness of 0.4 inch is entered at a position 3.0 inches down (x) and 2.0 inches across (y) from the current graphics REFERENCE position.

A.28 VLINE

Draw a vertical line.

Format:

VLINE [length]	
<u>Command Qualifiers</u>	<u>Default</u>
/P=Number	
/I=x,y	Current GRAPHIC POINTER position
/A=x,y	Current GRAPHIC POINTER position
/T=Thickness	Normal width dashed line

Prompts:

ENTER LENGTH: length

Command Parameters:

length -- Desired length of the line in inches. The prompt does not accept a carriage return since there is no default length for a line.

Description:

The VLINE command makes a request to draw (or plot) a vertical line. The length of the line is specified by the user.

The lines are inserted at the current GRAPHIC POINTER position.

Upon completion of the VLINE command, the GRAPHIC POINTER is left at the end of the line or the lower right corner of a thick line.

Options:

/P=n Place the line at the current FORMS LABEL POINT number "n" where "n" ranges between 1 and 12. This option is illegal when entered before the GETFRM command or after the ENDFRM command.

/I=x,y Place the line at an incremental distance x,y from the current GRAPHIC POINTER position.

/A=x,y Place the line at a distance x,y from the current REFERENCE and update the GRAPHIC POINTER position to this point.

/T=Thickness Specify the desired thickness of the line. Thickness can range from 0.1 inch to the limits of the page. If /T is not called, the line defaults to a thickness of one dot on the printer/plotter.

Examples:

1. CRT>VLINE/P=4/T=.2
ENTER LENGTH: 4

A line with a length of 4.0 inches and a thickness of 0.2 inch is entered at the current form's 4th LABEL POINT. (See the /P description above for restrictions.)

2. CRT>VLINE 3/I=1,2.5

A line with a length of 3.0 inches is entered at a position 1.0 inch down (x) and 2.5 inches across (y) from the current GRAPHIC POINTER position.

3. CRT>VLINE 2.5/A=3,2/T=4

A line with a length of 2.5 inches and a thickness of 0.4 inch is entered at a position 3.0 inches down (x) and 2.0 inches across (y) from the current graphics REFERENCE position.

A.29 VTAB

Move the REFERENCE ORIGIN vertically by a specified number of TAB increments.

Format:

VTAB [Vnum]

Prompts:

ENTER NUMBER OF VERTICAL TABS: Vnum

Command Parameters:

Vnum -- Number of vertical tabs to move the REFERENCE ORIGIN.

Description:

The VTAB command moves the REFERENCE ORIGIN by a multiple of the vertical tab distance (TABV; see SETTAB command) across the page. The distance between tabs (set by the SETTAB command) is checked to see that the limits of the page are not violated before moving the reference.

Options:

None

Examples:

1. CRT>VTAB
ENTER NUMBER OF VERTICAL TABS: 5
2. CRT>VTAB 5

EVFU	LXY PRINTER/PLOTTER's electronic vertical forms unit. This unit is set up with the page length (or form feed length) and is used to advance the LXY paper to the next physical page.
FORM BASE	A point on the FORM boundary (upper left corner) that is aligned to the REFERENCE ORIGIN at the time of form insertion (see GETFRM, Appendix A, Section A.9). All FORM GRAPHICS and LABEL POINTS are stored on the FORMS file with the position (X and Y) expressed relative to the FORM BASE on the form.
FORMS	A collection of GRAPHIC COMMANDS or REQUESTS that can be saved and later called to recreate a plotted output. Additional graphic data can be inserted into the form at the time of actual output.
GRAPHIC FIGURES	Any plotted figures generated by the BCPLIB routines (such as barcodes, block character, lines, etc.).
GRAPHIC POINTER	This is insertion position on the PLOT PAGE for graphic figures (such as barcodes, block characters, lines, etc.).
GRAPHIC REQUEST	<p>This is the stored (or buffered) data that occurs when the user calls on the BCP software to draw something such as:</p> <ul style="list-style-type: none">• To the system operator (or CREATE user). This is the result of entering a graphic figure command (BCODE, BLOCK, TEXT, VLINE, HLINE, VDASH, HDASH, DEFFRM, PUTFRM) or,• To the system programmer (or BCPLIB user). This is a result of calling one of the GRAPHIC SUBROUTINES.
LABEL POINTS	A set of offset points (defined at form creation time) defined relative to the FORM BASE. LABEL POINTS can later be referenced to position variable GRAPHIC FIGURES for insertion at the time of actual plot output.

PAGE (PLOT PAGE) The PAGE (or PLOT PAGE) is the external equivalent of the LXY PRINTER/PLOTTER page. The PAGE length and width is adjustable to match the PRINTER/PLOTTER page length and width.

REQUEST A general term for GRAPHIC REQUEST. This also includes all nongraphic figure CREATE commands and BCPLIB subroutine calls (such as MOVREF, MOVPTR, etc.).

SECTION A variable sized portion of the output page. The plot data may be output by a sequence of PAGE SECTIONS instead of whole PAGES (this is usually done when there are too many GRAPHIC REQUESTS to buffer for the entire PAGE).

APPENDIX C
ERROR MESSAGES

C.1 WARNINGS

- +2 REQUEST BUFFER NEARLY FULL -- This warning message occurs when five or fewer storage packets remain in the request pool.

C.2 ERROR

- 100 GRAPHIC FILE WRITE ERROR -- This error occurs when writing plot information on the GRAPHICS FILE. The REQUESTS for this SECTION are lost and the plot must be restarted. Check for device problems (such as disk full, device offline, etc.).
- 101 CANNOT OPEN GRAPHIC FILE -- This happens while trying to initialize or open a file for graphic output and an error condition occurs. It is impossible to continue with the current plot. Check for device problems (such as disk full, device off-line, illegal UIC, etc.).
- 102 GETFRM - ERROR CLOSING FORMS FILE -- The FORMS file following the contents of the defined form may be lost. Check for device problems (such as disk full, device off-line, etc.).
- 103 ILLEGAL ARGUMENT (PARAMETER) SPECIFIED -- One of the parameters supplied by the user is illegal (such as characters in a numeric field). Reenter the request.
- 104 GETFRM - CANNOT OPEN FORMS FILE -- The program cannot open the FORM file requested by the DEFFRM request. This happens if the form was not saved (SAVFRM) at the time of definition, if the form name was misspelled, or if device problems occurred (such as off-line, write locked device, etc.).
- 105 GETFRM - FORMS FILE READ ERROR -- This error occurs while reading the FORMS file as a result of the GETFRM request. Check for device problems (such as disk full, device off-line, etc.).
- 106 DEFFRM - CANNOT CREATE FORM FILE -- A FORM File was initialized (or opened) in response to a DEFFRM request and an error occurred. Check for device problems (such as illegal device, device off-line, etc.).

- 107 DEFFRM - FORMS FILE WRITE ERROR -- Error occurred while trying to write on the FORMS file following the DEFFRM request. The contents of the currently defined form may be lost. Check for device errors (such as write-locked device, device off-line, etc.).
- 108 SAVFRM - ERROR CLOSING FORMS FILE -- This error occurs while trying to close the FORMS file following the SAVFRM request. The contents of the currently defined form may be lost. Check for device problems (such as device off-line, write-locked device, etc.).
- 109 X LIMIT FAILURE (SECTION) -- The current request (TAB, MOVREF, GETFRM, or PUTFRM) defines a reference in the previous PAGE SECTION (or PAGE). If a PAGE SECTION or PAGE was processed, any further requests to draw graphic figures or part of a figure in that PAGE SECTION or PAGE cannot be accepted.
- 110 X OR Y LIMIT FAIL - OUTSIDE PAGE BOUNDARIES -- The X or Y parameter requested was outside the page boundaries. Type SHOW PAGE to view the current default page boundaries.
- 111 FORM DEFINITION - X OR Y OUTSIDE FORM BOUNDARIES -- This error occurs only during the forms definition mode. The current REQUEST (BCODE, BLOCK, TEXT, etc.) defines a figure whose area extends beyond the limits defined by the DEFFRM REQUEST.
- 112 X OR Y LIMIT FAIL - OUTSIDE REFERENCE BOUNDARIES -- The X or Y parameter requested was outside the reference boundaries. Type SHOW POS to view the current default REFERENCE ORIGIN.
- 113 Unused.
- 114 REQUEST BUFFER FULL -- No GRAPHIC REQUESTS can be accepted until Phase 2 processing is initiated. The current REQUEST was aborted. Use the PROCES, PAGE, REPEAT, FINISH (CREATE END) REQUEST to continue.
- 115 CANNOT CHANGE FIGURE PARAMETERS WHILE PLOT IN PROGRESS -- It is illegal to make a SETFIG request once plotting starts (this is following any graphic figure request such as BCODE, BLOCK, TEXT, etc.).
- 116 PUTFRM - NO FORM FIXED IN MEMORY - A form must be fixed in memory with GETFRM/F before PUTFRM can be requested.

- 117 GETFRM - FORM ALREADY FIXED IN MEMORY -- Once a form is fixed in memory the user cannot request another form (GETFRM), fixed or not fixed. The user must request ERASE FIX before another GETFRM REQUEST can be made.
- 118 SETPAG - X LIMIT BEYOND EVFU CAPABILITY -- The LXY PRINTER/PLOTTER EVFU facility has a maximum form length of 22.0 inches (in the X direction).
- 119 ILLEGAL REQUEST SEQUENCE -- The DEFFRM request cannot be made at this time because the system is already in the FORM definition mode or because plotting is already in progress.
- 121 SAVFRM - FORM NOT BEING DEFINED -- A form must be defined (DEFFRM) before it can be saved.
- 122 CANNOT PROCESS PLOT SECTION WHILE DEFINING A FORM -- The PLOT section cannot be processed (PROCES or REPEAT request) while a form is being defined.
- 123 Unused.
- 124 FORM INSERTION - X OR Y FORM LIMIT VIOLATION -- Limit violation occurred while in the form insertion mode (GETFRM by PUTFRM requests). The area covered by the defined figure (resulting from a GRAPHIC REQUEST such as BCODE, BLOCK, etc.) extends beyond the current FORM limits. The REQUEST is aborted. To continue, either make an ENDFRM REQUEST or reposition the figure.
- 125 Y LIMIT SPECIFIED IS BEYOND LXY CAPABILITY -- The Y parameter requested is beyond the 13.2 inches the LXY is capable of handling.
- 126 X LIMIT IS NOT A MULTIPLE OF 0.5 INCH -- The X parameter requested must be a multiple of 0.5 inch. PAGE or VTAB lengths (SETPAG) must be expressed in 0.5 inch increments.
- 127 PLOT NOT INITIALIZED -- The PLOT must be initialized before any requests can be entered. Use the CREATE NEWPLT command or the BCPLIB INIT subroutine to initialize the PLOT.
- 128 PLOTTING IN PROGRESS: CANNOT RESET PAGE PARAMETERS -- You cannot make a SETPAG REQUEST once plotting starts (this is following any graphic figure request such as BCODE, BLOCK, TEXT, etc.).

- 129 PLOT IS ALREADY INITIALIZED: CANNOT BE REINITIALIZED --
A PLOT was already initialized and is currently in use.
If a new plot file is needed, the current plot file must
be terminated by the CREATE user's END command or BCPLIB
user's FINISH request. After a FINISH (END) request, a
new plot can be initialized.
- 130 Unused.
- 131 REQUEST BUFFER EMPTY, NO OUTPUT POSSIBLE -- A PROCES or
REPEAT REQUEST was made, but the REQUEST BUFFERS were
empty.
- 132 REPEAT - TAB HEIGHT SMALLER THAN HEIGHT OF GRAPHICS --
The height requested with VTAB is smaller than the
height of the graphics to be repeated. The current VTAB
parameters can be viewed by typing SHOW PAG.
- 133 ILLEGAL USE OF GETFORM OR PUTFORM. MUST TERMINATE WITH
ENDFORM.
- 134 CANNOT USE "MOVREF" REQUEST DURING FORM INSERTION.
- 135 GETFORM - CANNOT FIX A FORM WHILE DEFINING A FORM --
During the form definition mode, a form cannot be fixed
in memory.
- 136 CREATE COMMAND ERROR
- 137 GETFORM - FIXED FORM DOES NOT CONTAIN GRAPHIC DATA
- 333 UNRECOGNIZED ERROR -- This can be caused by a program
"bug" or a memory error.

PART II

BCPLIB (GRAPHICS LIBRARY) USER'S GUIDE

1.1 GENERAL DESCRIPTION

The Barcode and Block Character Plotter (BCP) software package generates both barcode and block character printing on an LXY PRINTER/PLOTTER. Using an interactive user program and a library of graphic routines, the BCP software package can produce:

- CODE 39 barcode,
- Block characters,
- Horizontal and vertical lines,
- Horizontal and vertical bars (thick lines),
- Horizontal and vertical dash lines, and
- 0.1 inch text characters.

Typical uses include the creation of:

- Shipping and identification labels,
- Forms,
- Signs, and
- Inventory and accounting documents.

1.2 OVERVIEW

The BCP software package is made up of two components, each designed for a different user.

- CREATE is an interactive program for the nontechnical user. CREATE is built on the foundation provided by BCPLIB and gives the user an on-line graphics generation capability for use in the production environment.
- BCPLIB (BCP Graphic Utility Library) is available for the technical user or programmer experienced in writing computer software. BCPLIB is a library of software building blocks that serves as a foundation for the graphics applications. It provides graphic facilities for both user application software and the BCP CREATE Program.

There are two front-ends (or ways) into the BCP software (Figures 1-1 and 1-2). The front-ends are either through the:

- Interactive CREATE program, or
- Optional user application program.

Both of these programs are built upon BCPLIB, with each accepting commands from a system's terminal. These front-ends pass the command information on to BCPLIB. This command information is called the GRAPHIC REQUEST. GRAPHIC REQUESTS are processed by BCPLIB in two phases due to physical limitations placed on the program (such as LXY PRINTER/PLOTTER paper movement and computer memory).

Phase 1

During this phase, the GRAPHIC REQUESTS are accepted, evaluated, and stored (buffered) into a special area of the program's space in computer memory called the REQUEST BUFFER.

Phase 2

Phase 2 is entered as a result of a special request (or CREATE command) such as the PROCES request. During this phase, the requests in the buffer are processed into plottable information, which is then written or temporarily stored in the GRAPHICS FILE, and is later transferred to the LXY PRINTER/PLOTTER.

Phases 1 and 2 can be repeated alternately, until an entire plot (which may be more than one page of output) is completed.

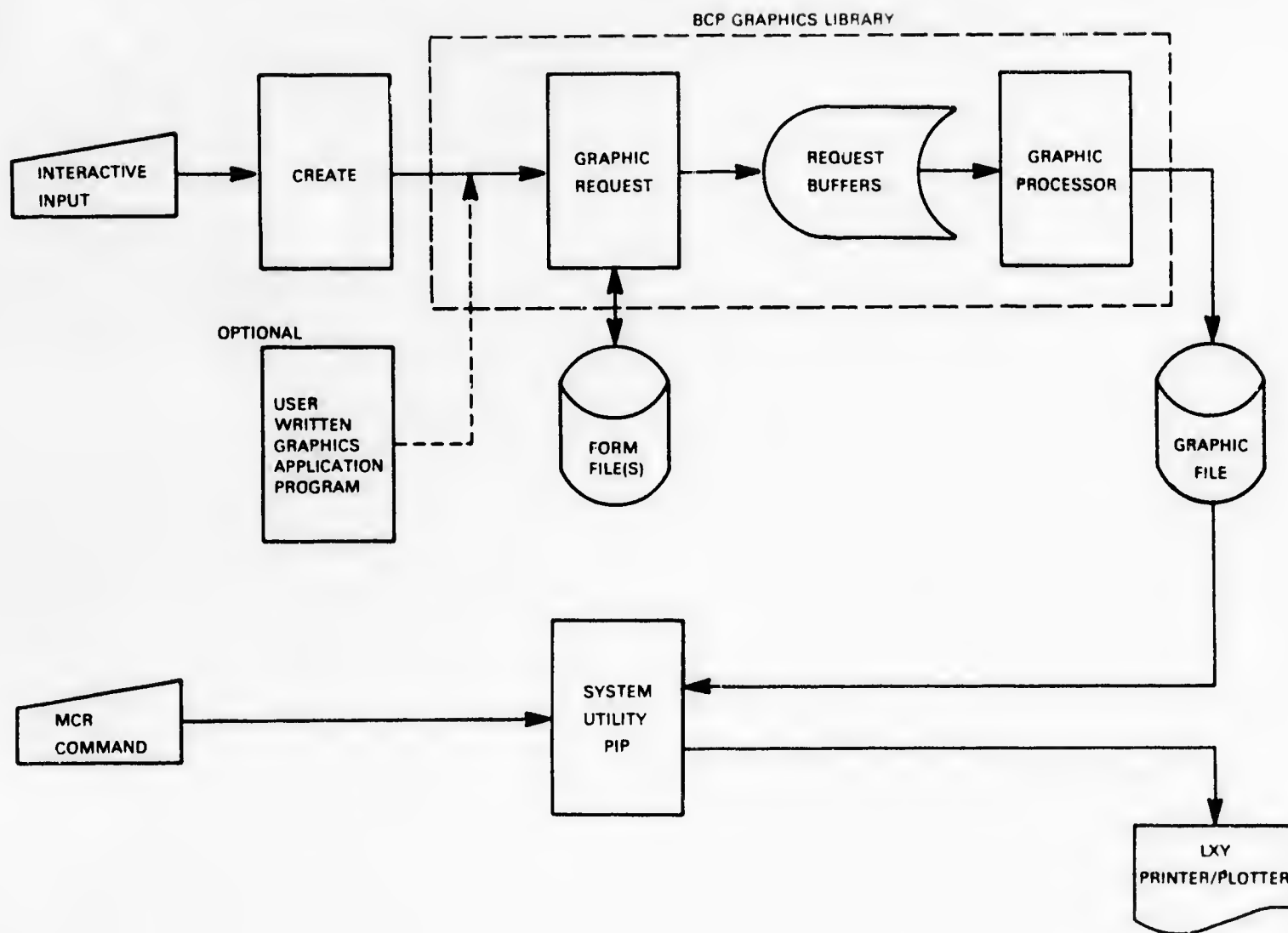
Once the plot is completed, the contents of the GRAPHICS FILE can be transferred directly to the LXY PRINTER/PLOTTER via the following methods.

- For RSX-11M -- Use the MCR utility PIP (or PRINT facility).
- For VAX/VMS -- Use the PRINT/NOFEED command.

1.3 BCP SOFTWARE CONCEPTS

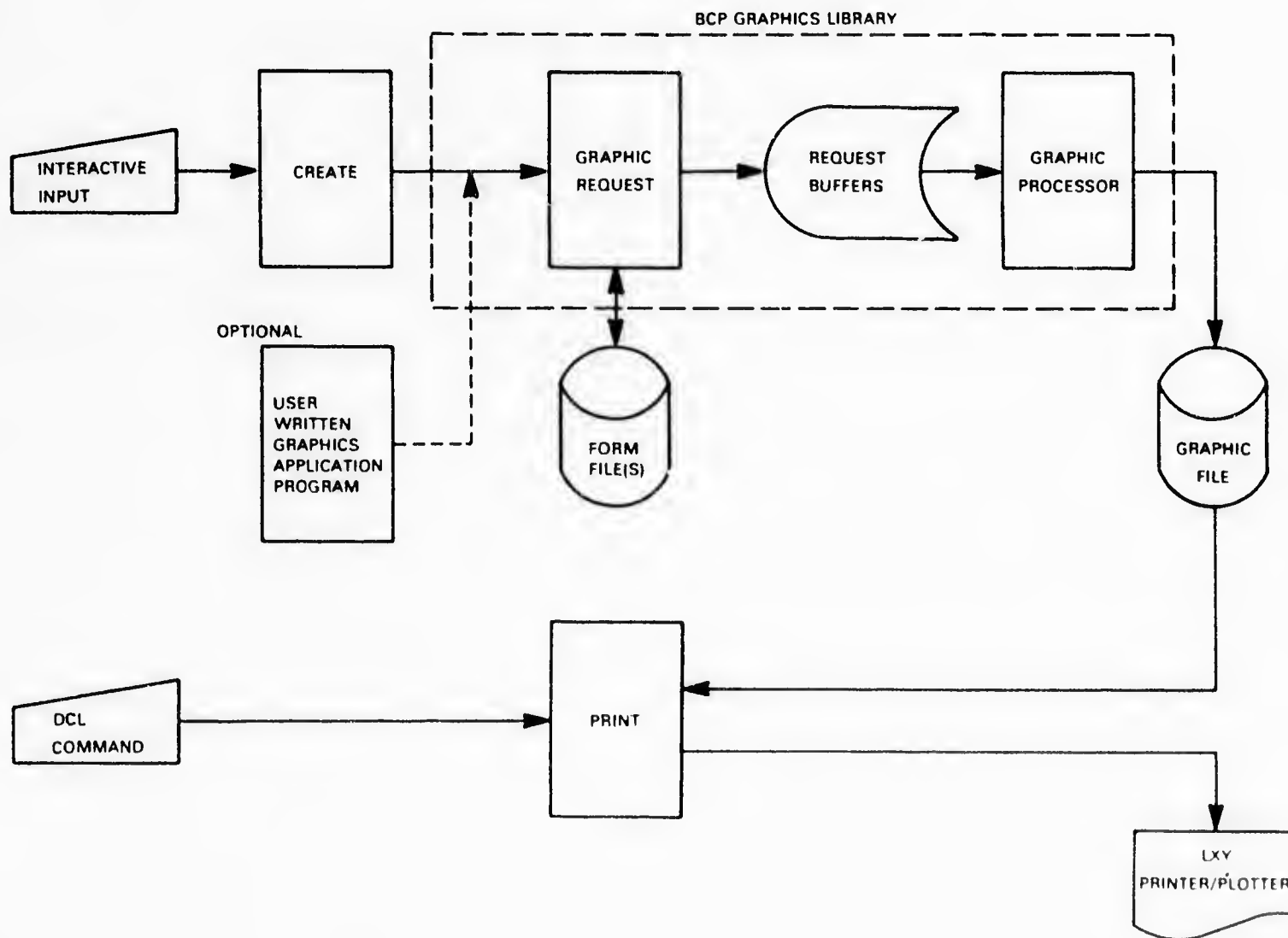
This section explains the concepts that must be understood in order to use the BCP software. They are:

- PLOT PAGE,
- SECTION,
- FORMS, and
- LABEL POINTS.



CS 2510

Figure 1-1 RSX-11M BCP System



CS 315A

Figure 1-2 VAX-11 BCP System

PLOT PAGE

The PLOT PAGE is the basic unit of plot output. It is an internal software equivalent of the LXY PRINTER/PLOTTER page. Since the PRINTER/PLOTTER paper can be of different lengths or widths, the PAGE boundaries can also be adjusted internally to match the PRINTER/PLOTTER PAGE SIZE. When GRAPHIC REQUESTS are received by the software, the dimensions of the figures are checked to see if they fit within the page boundaries. GRAPHIC (or plottable) information can be generated for more than one page, but each page is treated individually and terminated by a "TOP-OF-FORM" paper movement. The "TOP-OF-FORM" movement positions the paper to the top of the next physical page.

SECTION

If a lot of GRAPHIC REQUESTS are required to generate a plot, the REQUEST BUFFER could overflow before the output page is complete. To prevent this, Phase 2 of the GRAPHIC REQUEST may be triggered at any time and the requests received up to that time are processed. The area of the page covered by the graphic information in the REQUEST BUFFERS determines the page SECTION. Once a SECTION is processed, no more graphic input is accepted that has any part falling within the range of that SECTION. Following the output of that SECTION, a new SECTION is defined by updating the SECTION origin position to the start of the unused part of the page. The REQUEST BUFFERS have a capacity that can be exceeded by the number of GRAPHIC REQUESTS, graphic data should be entered in a top-to-bottom sequence. However, for most plots, the REQUEST BUFFER is adequate to store the graphics for the entire page. SECTIONS are defined and written as the result of the:

- PROCES request,
- END request, and
- REPEAT request.

FORMS

The BCP FORMS facility provides a means for the user to generate the fixed portion of a graphic output (such as labels and business forms) and store it for later use. This means that the fixed part of a graphic output must be generated only once. When producing the final output, the user can merge the fixed portion with the variable part of the graphic output. FORMS can be created during the Phase 1 process of the GRAPHIC REQUEST. Graphic requests defining the FORM are processed in the same way as a normal output. Phase 2 processing is bypassed and on the completion of the FORM, a special request is called to reformat the contents of the REQUEST BUFFERS and write the requests on the FORMS file.

Once a FORM is created, it can be repeatedly recalled during software production runs by referencing its name. FORMS are called in a similar manner to standard graphic figures such as barcodes. When a FORM is called, the FORM base is aligned with the REFERENCE ORIGIN (Section 1.4) position on input. The FORM data is read from the FORM file, reformatted, and stored in the REQUEST BUFFER like any standard figure. The variable portion of the FORM can be inserted at this time by making additional GRAPHIC REQUESTS.

A FORM can be used to define another, more complex FORM. A FORM can also be fixed into memory for repeated use. Not only does this save I/O activity on the FORM file, but a single image of the FORMS graphics may be stored for any number of different positions of the FORM on the plot page, thus saving REQUEST BUFFER space.

LABEL POINTS

LABEL POINTS are used as an aid in the positioning of the variable portion of FORMS during production runs. LABEL POINTS are a set of up to 12 predefined positions on the FORM which are selected by number (1 through 12) to position the variable graphics. The X and Y values associated with each LABEL POINT are offset (or incremental) distances from the FORM base. The FORM base is aligned with the REFERENCE ORIGIN (Section 1.4) when the FORM is requested and the LABEL POINTS then become GRAPHIC POINTER (Section 1.4) positions on the PLOT PAGE. By requesting a LABEL POINT by number, the user causes the GRAPHIC POINTER to be moved to the position indicated (by the LABEL POINT) and additional graphic input can then be added.

1.4 BCP SOFTWARE COORDINATE SYSTEM

To aid in the placement of the graphic figures on the PLOT PAGE (see Figures 1-3 and 1-4), the BCP software provides a coordinate system that is defined as:

- PAGE ORIGIN,
- SECTION ORIGIN,
- REFERENCE ORIGIN, and
- GRAPHIC POINTER.

PAGE ORIGIN

Upper left-hand corner of the physical line printer paper. This is plot (or absolute) position (0,0).

The ABSCISSA X is defined as being the vertical direction starting from the PAGE ORIGIN, increasing in value to the bottom of the page. Thus, the positive X direction corresponds with the paper movement and maintains a right-handed coordinate system. The resolution is 0.1 inch.

The ORDINATE Y is defined as being the horizontal direction starting from the PAGE ORIGIN, increasing in value to the right-hand side of the page. The resolution is 0.1 inch.

SECTION ORIGIN

This is the starting position of the current page section. The Y coordinate is always 0 and its X coordinate (XS) is an offset distance from the PAGE origin. The SECTION ORIGIN affects the user by becoming the lower limit for the X coordinate on his GRAPHIC REQUESTS. Initially, the SECTION ORIGIN starts at the PAGE origin which is 0,0. As sections are processed, it advances by the section height to define the start of the next section.

REFERENCE ORIGIN

This is a moveable coordinate system within the PAGE. Graphic figures are placed on the PAGE with respect to the REFERENCE ORIGIN (Xr,Yr). The X and Y position for the figures are offset distances from this point. The REFERENCE ORIGIN is moved by specifying new values for the Xr and Yr offsets from the PAGE origin. The REFERENCE ORIGIN is a mechanism to provide position independence (with respect to the PAGE) for the graphic figures on the PAGE. This is an important mechanism in processing FORMS and REPEAT requests.

The Xr value of the REFERENCE ORIGIN must not be smaller than the SECTION origin. As the PAGE sections are processed, the REFERENCE ORIGIN is advanced automatically with the SECTION origin.

GRAPHIC POINTER

Current position for graphic data entry is expressed as an offset Xc and Yc from the REFERENCE ORIGIN. Neither Xc nor Yc may take on negative values (for example, cannot be smaller than the REFERENCE ORIGIN).

1.5 GRAPHIC OUTPUT ON THE LXY PRINTER/PLOTTER

The LXY PRINTER/PLOTTERS place physical limitations on the BCP software operation. The PRINTER/PLOTTERS can only accept plottable data (raster data) one raster line (a line thickness of one dot, 0.0139 inch) at a time in a top-of-page to bottom-of-page and page-by-page sequence.

The LXY PRINTER/PLOTTER paper moves only in a forward direction, so that all graphic output must be coordinated with the printing movement of the paper. The plot data must be sorted and written on the PRINTER/PLOTTER one raster line at a time in a sequence acceptable to the PRINTER/PLOTTER. This places restrictions on the sequence of GRAPHIC REQUESTS entries by the user. As mentioned in Section 1.3, the BCP software sorts the stored requests according to physical position on the PAGE. Also, the BCP software can only accept graphic input in order according to PAGE. The restrictions to the order of graphic input on the PAGE depends on the number of requests and PAGE SECTIONS.

1.6 BCPLIB GRAPHICS PLACEMENT

1.6.1 Inserting Forms on the PLOT PAGE

FORMS are inserted on the PLOT PAGE at the current REFERENCE origin (Figure 1-3). The REFERENCE ORIGIN coincides with the FORM BASE. The FORM graphics are drawn from this point down the page (an increasing value of X) until the FORM length (FLIMX) is reached. In the same way, the FORM is drawn across the page until the FORM (FLIMY) width is reached.

The FORM LABEL POINTS, being offset values from the FORM BASE, now become GRAPHIC POINTER values.

The REFERENCE origin is moved in one of several ways. It may be moved:

- As the result of entering the TAB request (for example, VTAB or HTAB commands for the CREATE user or a call to the TAB subroutine for the BCPLIB user),
- As the result of entering the MOVREF request,
- As the result of entering the GETFRM and PUTFRM requests. These requests contain X and Y parameters to reposition the REFERENCE origin prior to the actual form insertion, or
- As the result of action taken at the completion of the PAGE, PRINT, and REPEAT requests. The REFERENCE ORIGIN is moved to the new PAGE or SECTION origin according to the request.

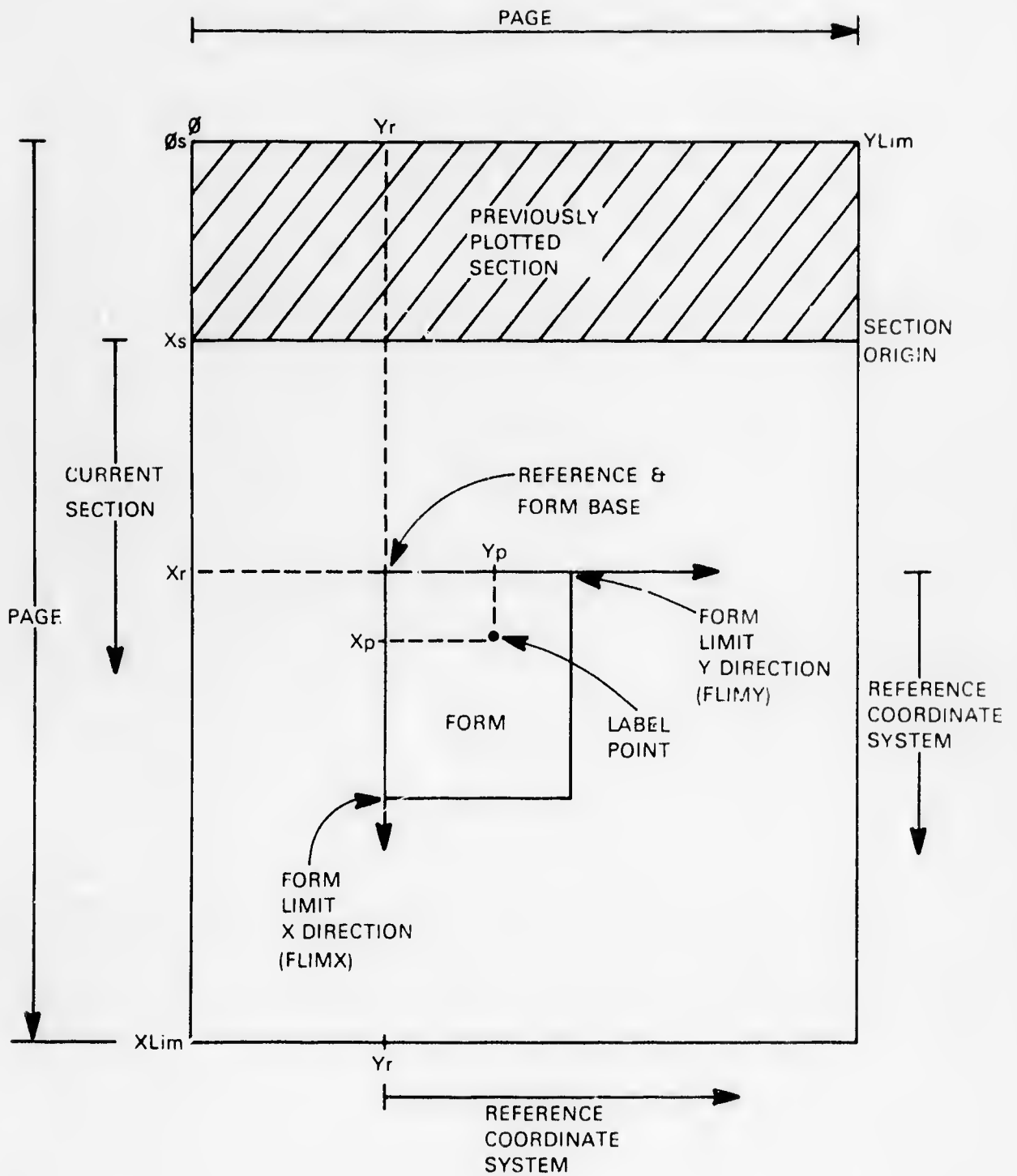
1.6.2 Inserting Graphic Figures on the PLOT PAGE

NOTE

Sufficient space (in both height and width) must be made available for the graphic figure that is to be drawn. The GRAPHIC POINTER position (Xc, Yc) must be chosen so that enough space in height and width is available to draw the graphic figure.

All graphic figures are inserted into the PLOT PAGE at the current GRAPHIC POINTER position. Figure 1-4 illustrates how the figure is placed using the block Character A as an example. The GRAPHIC POINTER (Xc, Yc) is assumed to be on the lower left corner of the figure, which is then drawn upward on the page (a decreasing X direction) until reaching the figure height and to the right by its width (SIZE).

The new position of the GRAPHIC POINTER is then set to the lower right corner figure plus an intercharacter space (Xce, Yce). The user should be cautioned that before placing the figure on the PLOT PAGE, the X values for the GRAPHIC POINTER position (Xc, Yc) must be at least as great as the figure height (SIZE).



CS 2539

Figure 1-3 FORMS

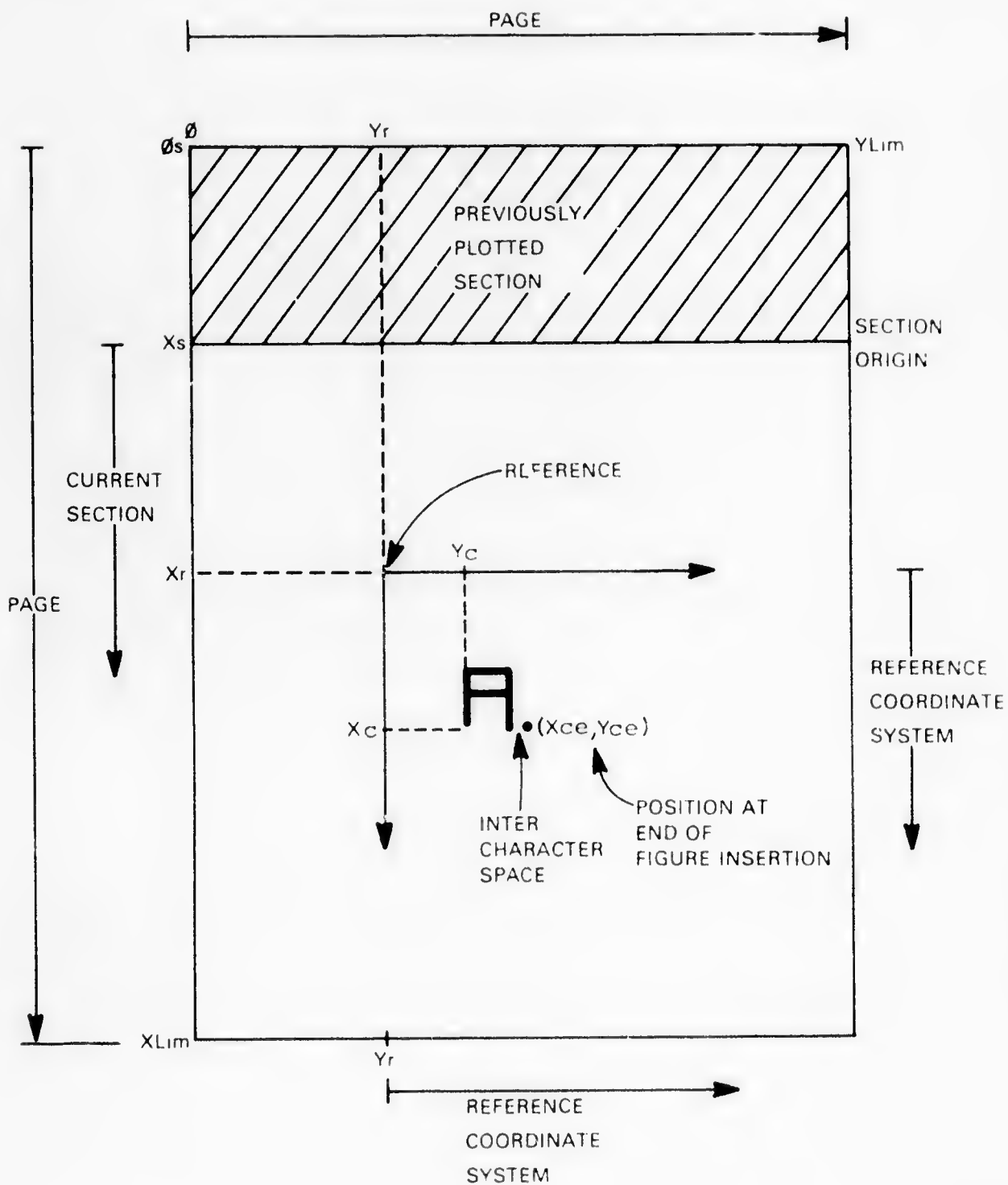


Figure 1-4 Graphics Insertion

For the LINES (VLINE, VDASH, HLINE, HDASH), the direction is implicit in the subroutine argument LENGTH which may have either a positive or negative value.

The GRAPHIC POINTER is moved in one of several ways. It may be moved:

- As the result of entering a MOVPTR request,
- As the result of entering a request to draw any graphic figure (BLOCK, BCODE, TEXT, etc.). Each graphic figure request includes parameters to reposition the GRAPHIC POINTER before actual figure insertion,
- As the result of any request that moves the REFERENCE ORIGIN (Section 1.4). Whenever the REFERENCE ORIGIN is moved, the GRAPHIC POINTER is reset to 0,0,
- As the result of action taken upon completion of the graphic figure requests (BLOCK, BCODE, etc.). BLOCK, BCODE, and TEXT leave the POINTER at the end of the character list plus the intercharacter space (1/6 of the character width), or
- As LINES (VLINE, HLINE, VDASH, HDASH) leave the pointer at the end of the line.

This chapter describes the use of the BCP utility (BCPLIB) facilities to generate graphic (plot) output from a user application program.

2.1 INTRODUCTION

The BCP utility library (BCPLIB) consists of graphic modules (subroutines) that provide a graphic foundation for the user's application program. These subroutines make up an objects library designed to be linked with the user application program at TASK build time.

The BCP library consists of three main parts.

- Preprocessing of Graphic Requests -- This is the first phase in the two-phase sequence required to process GRAPHIC REQUESTS. For Phase 1 the user application program makes the following types of requests (subroutine call statements).
 - Initialize the system for a new plot output and open the graphic's file.
 - Optionally change the default value of certain system parameters (such as page parameters, figure parameters, etc.)
 - Enter GRAPHIC REQUESTS such as LINES, BARCODE, BLOCK CHARACTERS, etc.
 - Optionally define forms.
- Tables and Buffers -- The final product from Phase 1 is the storage or queuing of the GRAPHIC REQUESTS collected so far for Phase 2 processing. The incoming GRAPHIC REQUESTS are verified for validity, reformatted, and then stored in special REQUEST BUFFERS for later Phase 2 processing.
- Postprocessing of the GRAPHIC REQUESTS -- The user's application program can trigger the Phase 2 or postprocessing activity by calling on one of the following routines.
 - PROCES
 - FINISH
 - PAGE
 - REPEAT

During Phase 2, the queued REQUESTS are processed by the graphic algorithms into plottable (rasterized) data, that is written on the GRAPHICS FILE. The plottable data is generated by the algorithms in the order of the LXY PRINTER/PLOTTER physical paper movement (using the plot mode facility). In this way, the graphic figures are generated simultaneously, according to their physical position on the output page, one raster line at a time. Later the GRAPHICS FILE can be directly transferred to the LXY PRINTER/PLOTTER via the system utility PIP (or PRINT FACILITY) for RSX-11M BCP or the PRINT/NOFEED command for VAX-11 BCP.

2.2 USING THE BCPLIB ROUTINES

Basically, there are four types of subroutines in the BCP library (refer to Appendix A for a list and description of these routines). See Figure 2-1 for overlaying possibilities for the BCPLIB routines (RSX-11M BCP only).

• Plot Control Routines

-- INIT/INI	Initialize for a new plot.
-- FINISH	Terminate the current plot.
-- PROCES	Process the current PAGE SECTION
-- REPEAT	Repeat the contents of the REQUEST BUFFERS.
-- SETPAG	Change the values for the default page parameters.
-- SETTAB	Change the values for the tabulation parameters.
-- ERASE	Erase or cancel queued requests.
-- EXAM	Examine specified system variables and parameters.

• Plot Positioning Routines

-- MOVPTR	Move the graphic pointer.
-- MOVREF	Move the reference origin.
-- PAGE	Advance to the next plot page.
-- TAB	Move the REFERENCE ORIGIN via tabs.

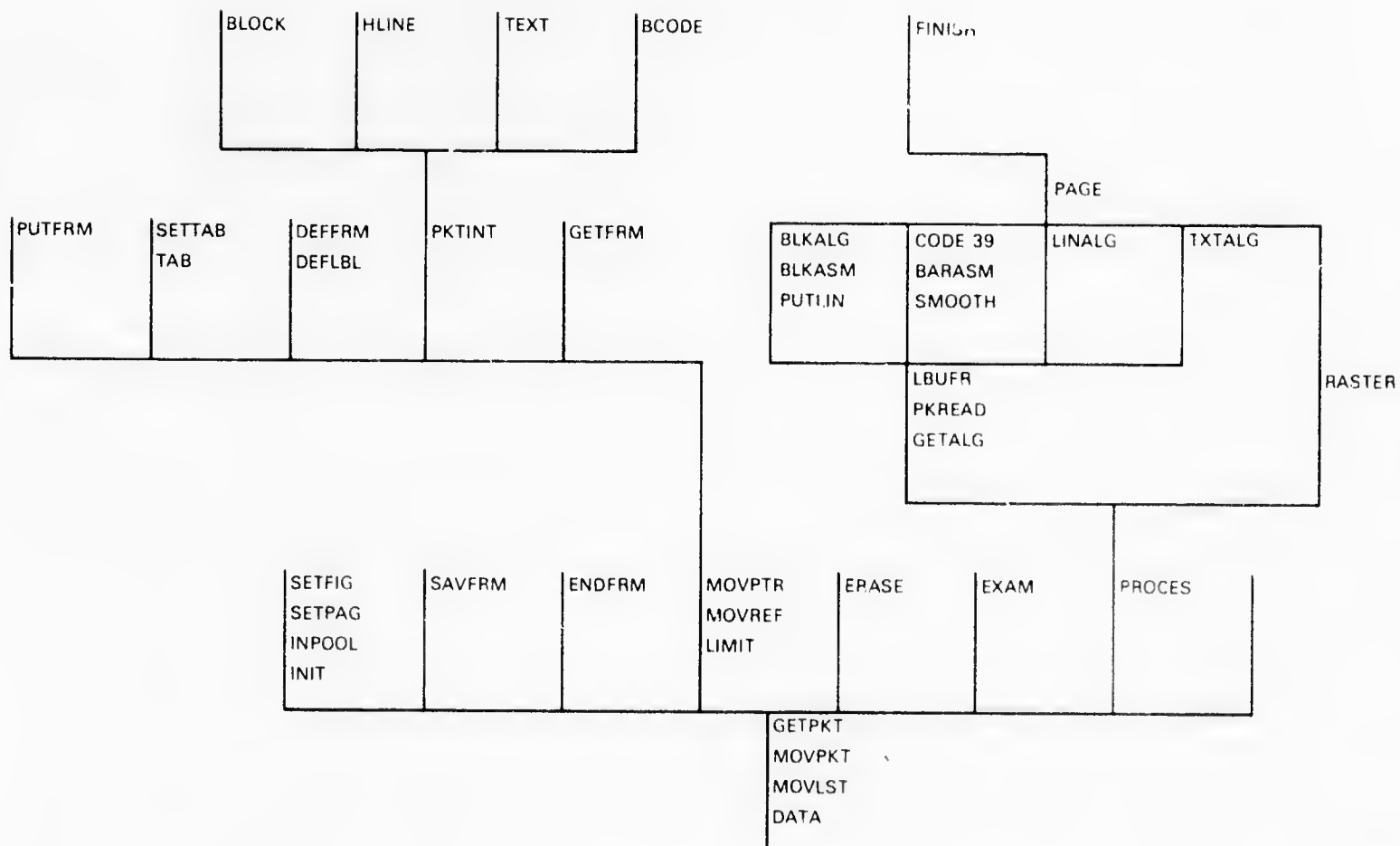


Figure 2-1 Overlay Guide (RSX-11M BCP)

- Graphic Figure Routines

-- BCODE	Draw a barcode.
-- BLOCK	Draw block character/s.
-- TEXT	Draw text.
-- HDASH	Draw a horizontal dash line.
-- HLINE	Draw a horizontal solid line.
-- VDASH	Draw a vertical dash line.
-- VLINE	Draw a vertical solid line.

- Form Routines

-- DEFFRM	Define a form.
-- SAVFRM	Terminate form definition.
-- DEFLBL	Define a form label point.
-- GETFRM	Insert a predefined form.
-- GETLBL	Get a form label point.
-- ENDFRM	Terminate form insertion.

2.2.1 Status Arguments

Most BCPLIB subroutine CALL statements include an argument called STATUS. STATUS returns to the user program a special code that describes the action taken as the result of the subroutine CALL (for example, request). The code has the following general characteristics.

- STATUS = +N Warning -- The request was honored. However, a situation exists that the caller should be aware of. The value of N identifies the situation (see Appendix C).
- STATUS = +1 Normal or successful return.
- STATUS = -N Error -- The error is identified by the numeric value of N. The request was aborted. Appendix C lists all error codes.

2.2.2 Format of Graphic Figure Routines

Each of the GRAPHIC ENTRY routines have three forms or variations. The variation specifies how the GRAPHIC FIGURE is to be positioned on the PLOT PAGE. Each GRAPHIC subroutine name consists of a basic five-character name and an optional one-character suffix. The suffix character (or lack of a suffix character) determines the variation.

- No Suffix Character -- Place the figure at an offset (incremental) distance from the current GRAPHIC POINTER position.

Example:

CALL BLOCK (N, LIST, DHT, DX, DY, SIZE, STATUS)

The block characters in the array list are to be drawn starting at an offset DX, DY from the current GRAPHIC POINTER position.

- P Suffix -- Place the figure/s at the specified FORM LABEL point.

Example:

CALL BLOCKP (N, LIST, DHT, P, SIZE, STATUS)

The argument "P" identifies the label point (1-12).

- A Suffix -- Move the GRAPHIC POINTER to a new position (X,Y) before placing the current figure.

Example:

CALL BLOCKA (N, LIST, DHT, X, Y, SIZE, STATUS)

The arguments X and Y specify the new position of the GRAPHIC POINTER, and hence, the insertion point for the characters in LIST.

2.3 GUIDELINES FOR USING BCPLIB TO GENERATE PLOTS

The following nine steps outline the sequence and general rules for PLOT generation using BCPLIB.

- 1a. Either to start a new plot -- Call the INIT subroutine.

Example:

CALL INIT ('BCP.PLT', 4, STATUS)

Comments:

All plots must start with a call to INIT.

To proceed from here:

Following the call to INIT, you may proceed with any step that follows, except for Steps 5 and 6.

- 1b. Or to initialize for FORM definition only -- Call the alternate entry INI.

Example:

CALL INI (STATUS)

Comments:

You may only define FORMS following this call.

2. To change the values of certain internal parameters -- Call one or more of the following routines.

- SETPAG -- To change the PAGE LIMITS, etc.
- SETFIG -- To change:
 - Default block character and barcode size;
 - Default barcode ratio flag;
 - Default block character, double-height flag; and/or
 - Default barcode text flag.
- SETTAB -- To change the values of the vertical and horizontal TAB distance.

Comments:

- These are optional subroutine calls.
- SETPAG and SETFIG can only be called at this time.
- SETTAB may be entered at any time following the call to INIT (or INI).
- SETTAB sets up the repeat intervals for the REPEAT routine.

To proceed from here:

Proceed to Step 3 or 4. If the subroutine call was made to SETTAB only, proceed to Step 7.

3. To define a form -- Enter the DEFFRM and DEFBL commands.

Example:

```
CALL DEFFRM ('FORM.FRM', 5.0, 6.2, STATUS)
CALL DEFBL (1, 1.0, 2.0, STATUS)
CALL DEFBL (2, 1.5, 2.5, STATUS)
```

o o o

```
CALL HLINE (5.0, 0.0, 0.0, 0., STATUS)
```

o o o

```
CALL BCODE (5, 'ABCDE', 0., 0., 0., STATUS)
```

o o o

```
CALL SAVERM (STATUS)          (See Step 6)
```

Comments:

- The call to DEFFRM is optional.
 - This step can only be preceded by Steps 1, 2, and 6.
 - DEFFRM forces the BCP software to enter a forms definition mode of operation in which:
 - Up to 12 label points can be defined by calls to the DEFLBL routine, and
 - Form graphics can be inserted via Step 4 routines below.
 - Although the GETFRM routine can be used while defining the form, its FIX option is illegal while in this mode.
 - The FORM definition process must be terminated by a call to SAVFRM (Step 6).
4. To enter graphic figures -- Call one or more of the following routines.

BLOCK	Draw BLOCK character/s.
BCODE	Draw CODE 39 barcode.
TFXT	Draw text.
VLIN	Draw a vertical line.
HLIN	Draw a horizontal line.
VDASH	Draw a vertical dash line.
HDASH	Draw a horizontal dash line.
GETFRM	Insert a predefined form.
PUTFRM	Insert a fixed form.

Comments:

When calling GETFRM:

- A FORM insertion mode of operation is entered where label points can be accessed (see P Suffix graphic subroutines).
- GETFRM or PUTFRM and hence FORM insertion mode is terminated by:
 - A call to the ENDFRM routine; and/or
 - A call to PRINT, PAGE, or REPEAT.
- During FORM definition, GETFRM cannot be called with the FIX option.

To proceed from here:

Proceed with any of the steps that follow. However, if GETFRM was entered, proceed with Step 5.

5. To terminate FORM insertion -- Call ENDFRM.

Example:

CALL ENDFRM

Comments:

- The FORM insertion mode of operation initiated by a call to the GETFRM routine is terminated by one of the following CALL statements.

Another GETFRM or PUTFRM
ENDFRM
PRINT
PAGE
REPEAT

- Following termination of the FORM insertion mode (normally by ENDFRM), the form's label points are no longer accessible. Therefore, the graphic figure routines called by using the P suffix (for example, BLOCKP, CODEP, and TEXTP) cannot be used.

To proceed from here:

If you are in the forms definition mode (DEFFRM), proceed with either Step 4 or 6.

If you are not in the forms definition mode, proceed with either Step 4, 7, 8, or 9.

6. To terminate form definition mode and save the form -- Call the SAVFRM routine.

Example:

CALL SAVFRM (STATUS)

Comments:

SAVFRM must be called to terminate FORM definition.

To proceed from here:

Proceed to either Step 2, 3, 4, or 9

7. To repeat the current contents of the REQUEST BUFFER -- Call the REPEAT subroutine.

Example:

CALL REPEAT (2,3,STATUS)

Comments:

REPEAT makes the following assumptions.

- The intervals between the graphics being repeated is determined by the values assigned to HTAB and VTAB (default values are 1.0 inch for each, the values may be changed by calling SETTAB).
- REPEAT acts upon the entire contents of the REQUEST BUFFER, not just the last graphic figure entered.
- Following REPEAT by a call to PRINT is unnecessary.

To proceed from here:

From here, you may go back as far as Step 4.

8. To terminate the current plot page -- Call the PAGE routine.

Example:

CALL PAGE (STATUS)

Comments:

PAGE causes the current contents of the REQUEST BUFFERS to process and output. Following this, LXY "TOP-OF-FORM" character is output to the printer. The plot is then reinitialized to the top of the next page.

To proceed from here:

From here, you may go back as far as Step 4.

9. To terminate the current plot -- Call either of these two routines: either EXIT or FINISH

Example:

CALL FINISH Terminate plot and wait for a NEWPLT command.

Comments:

None

2.4 COMPILING AND LINKING AN APPLICATION PROGRAM

After an application program is coded, it should be compiled. A typical compilation command string is shown below.

- For RSX-11M:
Example:

```
>F77 PROGRAM=PROGRAM.FTN
```

- For VAX/VMS:
Example:

```
$FOR PROGRAM
```

Once the program compiles free of errors, the following steps should be taken.

- For RSX-11M:
Task build (link) the program shown below.

Example:

```
>TKB  
TKB>PROGRAM/FP=PROGRAM,LB:[1,1]BCPLIB/LB,F4POTS/LB  
TKB>MAXBUF=140  
TKB>UNITS=8  
TKB>ASG=SY:7:8  
TKB>//
```

If you wish to use an overlay descriptor (ODL) file, see the file, DEMOBLD.CMD, for example of how this file can be built. Also, refer to the RSX-11M BCP Overlay Guide (Figure 2-1, page 2-3) for overlaying possibilities. Task build (link) as follows.

Example:

```
>TKB  
TKB>PROGRAM/FP=PROGRAM/MP  
TKB>MAXBUF=140  
TKB>UNITS=8  
TKB>ASG=SY:7:8  
TKB>//
```

- For VAX/VMS:
Link the program as shown below.

Example:

```
$LINK PROGRAM,BCPLIB/LIB/INC=BCDATA
```

The program can be run at this time.

Due to the variety of system configurations available to the user, this chapter only provides an outline for the system requirements and installation.

3.1 SYSTEM REQUIREMENTS

The general system requirements are as follows.

- A PDP-11 system running a mapped RSX-11M (V4.0) and equipped with a floating-point processor
- A minimum 28K partition to run the CREATE program
- 200 blocks of on-line disk storage for the GRAPHICS FILE
- FORTRAN 77 Compiler and Object Time System (OTS)
- An LXY Printer/Plotter

NOTE

During system generation of your RSX-11M V4.0 system, the LP11-Z or LP11-W controller, under peripheral options for the line printer, are recommended; for example, LPR Controller 0[D:200, 177514, LA180, 0, 300, 16130] [S]:,,LP-11-Z.

3.2 SYSTEM INSTALLATION

The outline of installation is as follows.

- MOUNT MEDIA -- Depending upon system configuration, this could involve a variety of media types.

Media Types

9-Track 800 BPI Magtape
RL02 Disk Cartridge
9-Track 1600 BPI Magtape
RL01 Disk Cartridge
RK06 Disk Cartridge
RX01 Floppy Diskette

The media label is BCP. An example of an RL01 distribution kit is:

>MOUNT DL0: BCP

- RESERVE DISK WORK AREA -- This work area is for the transfer of BCP files and for building the software package. Now set the default UIC to this area, for example:

```
>SET /UIC={200,200}
```

- PRINTER/PLOTTER -- If the PRINTER/PLOTTER is not set up for lower case characters and 134 characters/line, it can be changed by the MCR SET command, for example:

```
>SET /LOWER=LP0:
>SET /BUF=134.
```

- COPY BCP DISTRIBUTION FILES -- This is from the distribution media to the work area, for example:

```
-- Using RL01 distribution
```

```
>SET /UIC={200,200}
>PIP dev:=DL0:[200,200]*.*
```

```
-- Using magtape distribution
```

```
>FLX dev:/RS=MM0:[200,200]*.*/DO
```

From here two courses of action are possible. They are either to:

- Install the CREATE program, or
- Install only the BCPLIB library.

To install the CREATE program, you need to invoke a special command procedure file called CREATE. This file not only builds the CREATE program, but optionally builds the BCPLIB library.

Type: >@CREATE

Then respond to the command file's dialog as in the following example.

NOTE

The installer's responses are underlined. The installation takes approximately 20 minutes.

```

>@CREATE
>* HAVE YOU COMPILED THE CREATE PROGRAM? [Y/N]: N
>* ENTER THE NAME OF THE FORTRAN-77 TASK: [S]: F77
>F77 CREATE=CREATE.FTN
>MAC TPARS7=TPARS7
>LBR CREATE/CR:200:64:64:OBJ=CREATE,TPARS7
>* HAVE YOU BUILT THE BCP LIBRARY? [Y/N]: N
>F77 BCP=BCP.FTN
>F77 BCREQ=BCREQ.FTN
>F77 BCDATA=BCDATA.FTN
>F77 BCPRINT=BCPRINT.FTN
>MAC BARASM=BARASM.MAC
>LBR BCPLIB/CR:300:100.:100.:OBJ=BCDATA,BCP,BCREQ,BCPRINT,BARASM
>* HAVE YOU BUILT THE CREATE OVERLAY DISCRIPTOR FILE (ODL)? [Y/N]:N
>* ENTER THE NAME OF THE FORTRAN-77 OTS LIBRARY: [S]: F4POTS
>TKB @CRLINK
>@ <EOF>

```

NOTE

You may wish to install the CREATE.TSK image you just built on the system disk UIC [1,54] area for general use. Similarly, BCPLIB.OLB could be transferred to LB:[1,1] for general use.

To install only the BCPLIB library, use the BCPLIB command file. This command file builds the BCPLIB library from the BCP source files. To use BCPLIB.CMD, type:

```
>@BCPLIB
```

and respond to the command files dialog as in the following example. The installer's responses are underlined. The installation takes approximately 8 minutes.

```

>@BCPLIB
>* HAVE YOU BUILT THE BCP LIBRARY? [Y/N]: N
>* ENTER THE NAME OF THE FORTRAN-77 TASK: [S]: F77
>F77 BCP=BCP.FTN
>F77 BCREQ=BCREQ.FTN
>F77 BCDATA=BCDATA.FTN
>F77 BCPRINT=BCPRINT.FTN
>MAC BARASM=BARASM.MAC
>LBR BCPLIB/CR:300:100.:100.:OBJ=BCDATA,BCP,BCREQ,BCPRINT,BARASM
>@ <EOF>

```

NOTE

You may wish to move the BCPLIB.OLB file just generated to your system disk (logical device LB:) UIC [1,1] for general use.

4.1 SYSTEM REQUIREMENTS

- VAX/VMS (V3.2) or later
- 200 blocks of on-line disk storage for the GRAPHICS FILE.
- VAX-11 FORTRAN
- An LXY PRINTER/PLOTTER

4.2 SYSTEM INSTALLATION

The VAX-11 BCP distribution consists of:

- BCP User's Guide, and
- Two (2) floppy disks containing distribution files (VAX-11/780, VAX-11/782),
or
- Two (2) TU58 DECTAPE II cartridges containing distribution files (VAX-11/750, VAX-11/730).

The installation instructions for VAX-11 BCP are listed below.

1. The following privileges are needed for the installation.

- CMKRNL
- OPER
- SYSNAM
- SYSPRV
- LOG_IO

Example: SET PROC/PRIV=(CMKRNL,OPER,SYSNAM,SYSPRV,LOG_IO)

2. Set the UIC to [1,4].

Example: SET UIC [1,4]

3. Set the default as follows. SET DEF SYSS\$SYSROOT:[SYSUPD]
4. Mount the destination device if it is not presently mounted.
5. Create the destination directory if it does not exist.

Example: Create/dir DBC0:[BCP]

Note

The user must know the device name and number of the LXY PRINTER/PLOTTER.

6. Type @VMSUPDATE and respond to the dialog. VMSUPDATE calls the BCP installation command file.
 - The BCP installation command file:
 - Copies all files to the destination device and directory,
 - Compiles and links the library (BCPLIB) and CREATE programs,
 - Optionally compiles and links the BCP DEMO program, and
 - Optionally sets the LXY printer characteristics to those required by the BCP software - lowercase, printall, nowrap, and width=134.
 - A sample installation dialog is attached
7. The user is informed of the success or failure of the BCP installation. If the installation fails, recheck Steps 1 through 5. If the steps match the user installation, check for device problems.
8. Upon successful completion of the BCP installation, refer to Appendix D for BCP verification procedures.

The following command files provide the ability to rebuild any or all parts of VAX-11 BCP. They are:

- BCPLIB.COM -- Used to recompile and relink the BCPLIB routines,
- CREATE.COM -- Used to recompile and relink the CREATE program, and
- DEMO.COM -- Used to recompile and relink the DEMO program.

Sample Dialog:

```
$SET PROC/PRIV=(CMKRNL,OPER,SYSNAM,SYSPRV,LOG_IO)
$SET UIC [1,4]
$SET DEF SYS$$SYSROOT:[SYSUPD]
$@VMSUPDATE
```

VMS Update Procedure

This command procedure performs VAX/VMS software updates and optional software installations for VAX/VMS Release 3. During this sequence, the standard console medium is not present in the console drive. Therefore, the system may be vulnerable to a power failure or other fatal crash. If a system crash occurs during this period, the update sequence can be restarted at the beginning of the first incomplete update.

Dismount the current console medium.

Please place the first volume in the console drive

Are you ready to continue?: Y

%MOUNT-I-MOUNTED, BCP1 mounted on _CSA1

Starting installation of VAX-11 BCP

6-APR-1983 14:10:58

Your UIC must be set to [1,4] and you must have the following privileges to install BCP:

OPER	CMKRNL	LOG_IO
SYSPRV	SYSNAM	

Do you have these privileges and want to continue [Y/N]: Y

To print barcode output on an LXY PRINTER the following printer characteristics must be set:

lowercase	printall
nowrap	width=134

Do you wish to set the characteristics now: Y

Enter the LXY PRINTER device name and number "ddnn:" : LPA0:

Enter BCP destination area "device:[directory]" : DBC0:[BCP]

Remove the first volume (BCP1) and place the second volume (BCP2) in the drive.

6-APR-1983 14:16:57

Is the second volume (BCP2) in the drive [Y/N]: Y

6-APR-1983 14:16:57

%MOUNT-I-MOUNTED, BCP2 mounted on _CSA1:

Copying the CREATE HELP file into the system help area

*** The programs are now being compiled and linked ***

Do you wish to compile and link the demo program [Y/N]: Y

BCP installation has completed successfully --

- * all files have been copied
- * all required programs are compiled and linked

6-APR-1983 14:25:36

Are there more kits to process?: N

Please place the system console medium in the console drive.

Are you ready to continue?: Y

%MOUNT-I-MOUNTED, CONSOLE mounted on _CSA1:

Requested update sequence is complete.

Appendix A is an alphabetical listing of the command subroutines.

A.1 BCODE

Draw a BARCODE of the characters in the array string.

Format:

```
BCODE (N, LIST, DX, DY, SIZE, STATUS)
```

```
BCODEA (N, LIST, X, Y, SIZE, STATUS)
```

```
BCODEP (N, LIST, P, SIZE, STATUS)
```

Arguments:

- N An integer*2 variable or array element that is the number of characters in LIST below.
- LIST A character variable or constant containing the characters being plotted in barcode format.
- DX,DY Real variables or array elements that specify an offset distance in inches (resolution of 0.1 inch) to move the GRAPHIC POINTER for insertion of the barcode figure.
- X,Y Real variables or array elements that specify a new position for the GRAPHIC POINTER for insertion of the barcode figure. The units are inches and resolution is 0.1 inch.
- P An integer*2 variable or array element that specifies a form LABEL POINT. The GRAPHIC POINTER is to be set to the value of the named LABEL POINT for insertion of the barcode figure. P is a number between 1 and 12.
- SIZE A real variable or array element that specifies barcode size (height) in inches. The resolution is 0.1 inch.

STATUS An integer*2 variable or array element where the BCODE subroutine stores a status code describing the action taken on the CALL (or request). On normal or successful return from the subroutine, STATUS has a value of 1. Otherwise, either an error or warning condition results. Refer to Appendix C for a description of this problem.

Description:

The BCODE subroutine makes a request to draw (or plot) a list of characters in CODE 39 BARCODE format. The barcode height is determined by the argument SIZE (unless its value is 0 in which case an internal parameter, by the same name, is used). Initially, the internal (or default) value for SIZE is 1.0 inch, but this quantity can be changed by a call to the SETFIG subroutine.

The barcode is drawn with a ratio (wide line widths to thin line widths and similar spaces) of either 2:1 or 3:1, that is dependent upon the value of an internal parameter RATIO. RATIO has an initial value of 2, however, the SETFIG command can be used to change this value to 3.

Barcode text optionally can be printed along with the barcode figure. The barcode height is shortened by 0.2 inch for text insertion with this option. Initially, the text option is set to include the printed text, however, the SETFIG command may be used to specify that no text be included (see the OMIT parameter) when plotting barcodes.

The barcode is inserted at the current GRAPHIC POINTER position starting at the lower left corner of the barcode figure itself.

Upon completion of the BCODE command, the GRAPHIC POINTER is left at the lower right corner of the figure.

Examples:

1. CALL BCODE (4, 'ABCD', 6.0, 1.0, 5.0, STATUS). Generate a CODE 39 barcode for the characters "ABCD". The barcode is drawn at a height of 5.0 inches and inserted in the PLOT PAGE at a position 6.0 inches down the page (X direction) and 1.0 inch across the page (Y direction) from the current GRAPHIC POINTER position.
2. CALL BCODEP (5, 'A123Z', 5, 0, STATUS). Generates a CODE 39 barcode for the characters "A123Z" at the position defined by the current form's 5th LABEL POINT.

A.2 BLOCK

Draw block characters listed in the array list.

Format:

```
BLOCK (N, LIST, DBHT, DX, DY, SIZE, STATUS)
```

```
BLOCKA (N, LIST, DBHT, X, Y, SIZE, STATUS)
```

```
BLOCKP (N, LIST, DBHT, P, SIZE, STATUS)
```

Arguments:

- N** An integer*2 variable or array element that is the number of characters in LIST below.
- LIST** A character variable or constant containing the characters to be plotted.
- Legal characters are: 0-9
A-Z
\$, '*-/,:;=?%
- DBHT** Logical constant or variable. IF DBHT has the value of .TRUE., then draw the characters double height. If DBHT has any other value, refer to the system parameter by the same name to determine how to draw the characters. The initial value for the default system parameter DBHT is .FALSE. See SETFIG command to change the default value.
- DX,DY** Real variables or array elements that specify distance in inches (resolution of 0.1 inch) to move the GRAPHIC POINTER for insertion of the block characters.
- X,Y** Real variables or array elements that specify a new position for the GRAPHIC POINTER for insertion of the BLOCK characters. The units are inches and resolution is 0.1 inch.
- P** An integer*2 variable or array element that specifies a form LABEL POINT. The GRAPHIC POINTER is set to the value of the named LABEL POINT for insertion of the BLOCK characters. P is a number between 1 and 12.
- SIZE** A real variable or array element that specifies character size (height and width) in inches. The resolution is 0.1 inch.

STATUS An integer*2 variable or array element where the BLOCK subroutine stores a status code describing the action taken on the CALL (or request). On normal or successful return from the subroutine, STATUS has a value of 1. Otherwise, either an error or warning condition results. Refer to Appendix C for a description of this problem.

Description:

The BLOCK subroutine makes a request to draw (or plot) a list of characters in block format. Block formatted characters are drawn so that the character height and the width are the same unless the double-height option (/D) is specified. For double height, the character height is twice the width. The character width includes intercharacter spacing on the character's right side, that is 1/6th of the total width.

Whenever the size argument is 0, the character height and width is determined by the internal parameter by the same name (SIZE). Initially, SIZE has the value of 1.0 inch, however, its value can be changed by using the SETFIG subroutine.

The characters are inserted at the current GRAPHIC POINTER position starting at the lower left corner of the first character.

Upon completion of the BLOCK subroutine, the GRAPHIC POINTER is left at the last character's lower right corner (including the intercharacter space).

Examples:

1. CALL BLOCK (3, 'ABC',.TRUE.,1.0,1.0,0.,STATUS). Draw the three characters "ABC" in double-height block format. Move the GRAPHIC POINTER a distance of 1.0 inch horizontally and 1.0 inch vertically before inserting the characters into the PLOT PAGE. Since SIZE is 0, draw the characters having a size as defined by the system default SIZE parameter (see SETFIG subroutine).
2. CALL BLOCK (7, '123 ABC', 0.0,0.0,3.0,STATUS). Draw the seven characters "123 ABC" in standard (no double-height) block format. Insert the characters at the current GRAPHIC POINTER position. Draw the characters of height and width of 3.0 inches.
3. CALL BLOCKP (3, 'ABC',.false.,5,1.5 STATUS). Draw the three characters "ABC" in a standard block format with a SIZE of 1.5 inches. Insert the characters at the current form's 5th LABEL POINT.

A.3 DEFFRM

Define a form. Following the "DEFFRM" call, the user is expected to provide the form's graphics via standard subroutine calls in the identical manner as normal graphic input. When all the graphics are specified, the "SAVFRM" call is made to terminate the form definition and save the form for future use. All form graphic figures are checked to fall within the form boundaries (XLIM,YLIM).

Format:

```
-----  
|                                     |  
| DEFFRM (FILESP, UNIT, XLIM, YLIM, STATUS) |  
|                                     |  
|-----|
```

Arguments:

FILESP	Character constant or variable -- File name of the FORM FILE being created to store the new form. This is a FILE SPECIFICATION as defined in the <u>RSX-11 M/M-Plus MCR Operations Manual, Section 2.2, for RSX-11 BCP or VAX/VMS Command Language User's Guide</u> , Chapter 2, for VAX-11 BCP.
UNIT	Integer*2 -- If UNIT is not 0, then it becomes the form file's Logical Unit Number (LUN). If UNIT is 0, then the default LUN 8 is used.
XLIM	Real variable -- Maximum length of the form in inches. RESOLUTION is 0.1 inch.
YLIM	Real variable -- Maximum width of form in inches. RESOLUTION is 0.1 inch.
STATUS	STATUS is an integer*2 variable or array element where the DEFFRM subroutine stores a status code describing the action taken on the CALL (or request). On normal or successful return from the subroutine, STATUS has a value of 1. Otherwise, either an error or warning condition results. Refer to Appendix C for a description of this problem.

Description:

The DEFFRM subroutine initiates a FORMS definition mode of operation where the user can call graphic figure routines to describe the form in terms of lines, barcodes, text, and block characters. These calls are then saved for later use on a disk file. The FORM can also include LABEL POINTS defined by the DEFPTNT subroutine. When the FORM is completed, the user calls the SAVFRM routine to save the FORM and its contents on a special disk file created for this purpose. FORMS may not be defined after normal plotting starts.

Examples:

1. CALL DEFFRM ('NEWFRM.FRM', 0,5.0,6.0, STATUS) defines a form called NEWFRM.FRM and initiates a FORM definition mode of operation where the form's GRAPHICS are entered. A file called NEWFRM.FRM is created and the FORM header inserted. Such routines as BCODE, BLOCK, etc., can now be called to describe the FORM'S contents. A later call to SAVFRM terminates FORM building and stores the graphics on the disk file. The file is then closed and the form insertion mode is terminated.
2. CALL DEFFRM ('BCP.FRM',0,5.0,5.0, STATUS) defines a form to store on the file BCP.FRM. Associate logical unit 8 to the form file. The form's vertical limit is 5.0 inches and width is 5.0 inches.

A.4 DEFLBL

DEFINE a label point at the FORM offset position X,Y.

Format:

```
-----  
|                                     |  
|               DEFLBL (N,X,Y, STATUS)               |  
|                                     |  
|-----|
```

Arguments:

N	An integer*2 constant variable identifying the LABEL POINT number.
X,Y	Real variables that specify position on the FORM relative to the where the LABEL POINT is located. X and Y are expressed in whole or tenth inch units.
STATUS	An integer*2 variable or array element where the DEFLBL subroutine stores a status code describing the action taken on the CALL (or request). On normal or successful return from the subroutine, STATUS has a value of 1. Otherwise, either an error or warning condition results. Refer to Appendix C for a description of this problem.

Description:

DEFLBL defines a form LABEL POINT (Section 1.3). This subroutine may only be called while BCPLIB is in the FORMS definition mode of operation. The forms definition mode is initiated by a call to the DEFFRM subroutine and terminated by calling SAVFRM. DEFLBL associates a position X,Y relative to the form base and associates it to a LABEL POINT number. Later, this same relative position may be accessed via reference to the associated number.

A.5 ENDFRM

Terminate form definition. Close the form file.

Format:

```
-----  
|                                     |  
|                               ENDFRM                               |  
|                                     |  
-----
```

Arguments:

None

Description:

ENDFRM terminates the form insertion mode of operation initiated by calling GETFRM or PUTFRM subroutines.

z

A.6 ERASE

Erase current contents of the request buffers.

Format:

```
-----  
|                                     |  
|                               ERASE (ID, STATUS)                               |  
|                                     |  
|-----|
```

Arguments:

ID Character string or variable that identifies what to erase. ID can have a value of one of the three following character strings.

- REQ -- Erase REQUEST BUFFER only.
- FIX -- Erase FIXED FORM only.
- ALL -- Erase both buffers.

STATUS An integer*2 variable or array element where the ERASE subroutine stores a status code describing the action taken on the CALL (or request). On normal or successful return from the subroutine, STATUS has a value of 1. Otherwise, either an error or warning condition results. Refer to Appendix C for a description of this problem.

Description:

Enables the caller to erase the current contents of the REQUEST BUFFERS and/or FIXED FORM BUFFER. A call to ERASE with the ID argument having a value of FIX is the only way to remove a FIXED FORM from memory, short of terminating the PLOT.

A.7 EXAM

Format:

```
-----  
|                                     |  
|               EXAM (TYPE, BUFFER, STATUS)               |  
|                                     |  
|-----|
```

Arguments:

- TYPE** Character constant or variable identifying the type of parameters to examine. They are:
- POS -- POSITION parameters;
 - PAG -- PAGE parameters;
 - FIG -- FIGURES parameters;
 - FOR -- FORM parameters; and
 - ALL -- POS, PAG, FIG, and FOR parameters.
- BUFFER** Real array to contain the requested parameters. The contents depend upon parameter type(s) requested.
- POS -- 5 words to hold position data are the:
 - XR - X coordinate of REFERENCE ORIGIN,
 - YR - Y coordinate of REFERENCE ORIGIN,
 - XC - X coordinate of GRAPHIC POINTER,
 - YC - Y coordinate of GRAPHIC POINTER, and
 - SECTION - Start of current SECTION.
 - PAG -- 4 words to contain page parameters. The page parameters are:
 - XLIM - Page limit in X direction (LENGTH),
 - YLIM - Page limit in Y direction (WIDTH),
 - VTAB - Current value of the vertical TAB, and
 - HTAB - Current value of the horizontal TAB.
 - FIG -- 3 words that contain figure data are:
 - SIZE - Character size,
 - DBHT - Double-height FLAG (1 = single height; 2 = double height),
 - RATIO - Barcode ratio, and
 - BCTEXT - Barcode text FLAG (3 = include text in the barcode figure; N = omit text in the barcode figure).

- FOR -- 26 words to contain (GETFRM) form data. The form parameters are:
 - XLIM - Form length,
 - YLIM - Form width, and
 - POINT - 24 words of X,Y data defining label points in order - 2 words per label point.
- ALL -- 38 words to contain all of the above information in order shown.

STATUS

An integer*2 variable or array element where the EXAM subroutine stores a status code describing the action taken on the CALL (or request). On normal or successful return from the subroutine, STATUS has a value of 1. Otherwise, either an error or warning condition results. Refer to Appendix C for a description of this problem.

Description:

EXAM returns to the caller the current values of selected internal parameters. All parameters are converted to real format upon return, therefore, the logical variables should be interpreted as .FALSE. if the returned value is 0, and .TRUE. otherwise.

A.8 FINISH

FINISH terminates graphics and closes all files.

Format:

FINISH (STATUS)

Arguments:

STATUS An integer*2 variable or array element where the FINISH subroutine stores a status code describing the action taken on the CALL (or request). On normal or successful return from the subroutine, STATUS has a value of 1. Otherwise, either an error or warning condition results. Refer to Appendix C for a description of this problem.

Description:

Terminate the current plot and make preparations to receive another INIT subroutine call to start a new plot. If the request buffers are not empty, call the PROCES and PAGE routines as needed to complete the current plot and advance the page in preparation for a new plot to start.

A.9 GETFRM

Get FORM. Read FORM from file and enter contents into the REQUEST BUFFERS.

Format:

```
-----  
|                                     |  
|          GETFRM (FMFILE, FIX, X, Y, STATUS)          |  
|                                     |  
|-----|
```

Arguments:

FMFILE	A character constant or variable for the FORM file name. FMFILE is a file specification as defined in the <u>RSX-11M/M-Plus MCR Operations Manual</u> , Section 2.2 or <u>VAX/VMS Command Language User's Guide</u> , Chapter 2. A default of .DAT is used when the file extension is omitted.
FIX	Logical*2 flag. Set to .TRUE. to fix the FORM into memory.
X,Y	Real variables -- New position for the REFERENCE ORIGIN, and thus the FORM insertion point.
STATUS	An integer*2 variable or array element where the GETFRM subroutine stores a status code describing the action taken on the CALL (or request). On normal or successful return from the subroutine, STATUS has a value of 1. Otherwise, either an error or warning condition results. Refer to Appendix C for a description of this problem.

Description:

GETFRM requests a form from a disk file and inserts it in the REQUEST buffer in the same manner as a normal graphic request (such as BCODE) stores its data. In addition, GETFRM initiates a forms insertion mode of operation that permits accessing of LABEL POINTS by the graphic subroutines via the P suffix entry (see BCODEP, BLOCKP, TEXTP, etc., command descriptions).

The forms insertion mode is terminated by one of the following routines.

- Another GETFRM subroutine call
- ENDFRM
- PROCES
- PAGE
- REPEAT

The FIX argument may be used to cause the form contents being saved or fixed in a special place in the memory for shared use. A fixed form may be reused for any number of positions on the page via the PUTFRM command from a single copy of its graphic contents stored in this way.

Whenever a form is fixed into memory, another call to GETFRM cannot be made until the form is removed. A fixed form is removed only by the ERASE command using the FIX parameter.

Examples:

1. CALL GETFRM ('BINLBL.FRM',0,1.0,3.2,STATUS) reads a form from the file BINLBL.FRM and stores its graphic content into the REQUEST BUFFERS for later Phase 2 processing into plottable data. The FORM contents are inserted on the PLOT page at coordinate position (1.0,3.2).
2. CALL GETFRM ('PRICE.LBL',.TRUE.,3.6,5.2,STATUS) reads a form from the file PRICE.LBL and stores it into the fixed form buffer to FIX the form. The form is inserted on the PLOT page at coordinate position (3.6,5.2) and wherever specified by later PUTFRM subroutine calls.

A.10 GETLBL

Set current position pointer to label point N.

Format:

```
-----  
|                                     |  
|                               GETLBL (N, STATUS)                               |  
|                                     |  
|-----|
```

Arguments:

- N An integer*2 variable, constant, or array element specifying the LABEL POINT by number. N is any number between 1 and 12.
- STATUS An integer*2 variable or array element where the GETLBL subroutine stores a status code describing the action taken on the CALL (or request). On normal or successful return from the subroutine, STATUS has a value of 1. Otherwise, either an error or warning condition results. Refer to Appendix C for a description of this problem.

Description:

GETLBL moves the GRAPHIC POINTER to a new position that is associated with the current form's LABEL POINT N. This subroutine may only be called when the software is in the forms insertion mode of operation. The forms insertion mode is initiated by the GETFRM and PUTFRM subroutines and terminated by a call to the ENDFRM subroutine or any routine that initiates Phase 2 processing (PROCES, PAGE, REPEAT).

A.11 HDASH

Dashed horizontal line.

Format:

```
HDASH (LENGTH, DX, DY, THICK, STATUS)
HDASHA (LENGTH, X, Y, THICK, STATUS)
HDASHP (LENGTH, P, THICK, STATUS)
```

Arguments:

LENGTH	A real variable or array element that specifies the LINE LENGTH. The UNITS are in inches and the resolution is 0.1 inch.
DX,DY	Real variables or array elements that specify an offset distance in inches (resolution of 0.1 inch) moves the GRAPHIC POINTER for insertion of the DASHED LINE.
X,Y	Real variables or array elements that specify a new position for the GRAPHIC POINTER for insertion of the DASHED LINE. The units are inches and resolution is 0.1 inch.
P	An integer*2 variable or array element that specifies a form LABEL POINT. The GRAPHIC POINTER is set to the value of the named LABEL POINT for insertion of the DASHED LINE. P is a number from 1 to 12.
THICK	A real variable or array element that specifies the line thickness. If THICK has a value of 0, a normal line of a thickness of 0.039 inch is drawn. If THICK is greater than 0, then it specifies the thickness of the line in whole and/or tenth of an inch units.
STATUS	An integer*2 variable or array element where the HDASH subroutine stores a status code describing the action taken on the CALL (or request). On normal or successful return from the subroutine, STATUS has a value of 1. Otherwise, either an error or warning condition results. Refer to Appendix C for a description of this problem.

Description:

The HDASH routine makes a request to draw (or plot) a horizontal, dashed line. Length of the line is specified by the user.

The lines are inserted at the current GRAPHIC POINTER position.

Upon completion of the HDASH routine, the GRAPHIC POINTER is left at the end of the line or the lower right corner of a thick line.

A.12 HLINE

Format:

```
-----  
|                                     |  
|          HLINE (LENGTH, DX, DY, THICK, STATUS) |  
|          HLINEA (LENGTH, X, Y, THICK, STATUS) |  
|          HLINEP (LENGTH, P, THICK, STATUS) |  
|                                     |  
|-----|
```

Arguments:

LENGTH	Line length in inches with resolution of 0.1 inch (real variable).
DX,DY	Real variables or array elements that specify an offset distance in inches (resolution of 0.1 inch) moves the GRAPHIC POINTER for insertion of the LINE.
X,Y	Real variables or array elements specify a new position for the GRAPHIC POINTER for insertion of the LINE. The units are inches and resolution is 0.1 inch.
P	An integer*2 variable or array element specifies a form LABEL POINT. The GRAPHIC POINTER is set to the value of the named LABEL POINT for insertion of the line. P is a number between 1 and 12.
THICK	A real variable or constant that specifies the line thickness. If THICK has a value of 0, a normal line of thickness of approximately 0.039 inch is drawn. If THICK is greater than 0, a line of the specified thickness is drawn. The thickness is expressed in whole and/or tenth of an inch units.
STATUS	An integer*2 variable or array element where the HLINE subroutine stores a status code describing the action taken on the CALL (or request). On normal or successful return from the subroutine, STATUS has a value of 1. Otherwise, either an error or warning condition results. Refer to Appendix C for a description of this problem.

Description:

The HLINE command makes a request to draw (or plot), a horizontal line. Length of the line is specified by the user.

The lines are inserted at the current GRAPHIC POINTER position.

Upon completion of the HLINE command, the GRAPHIC POINTER is left at the end of the line or at the lower right corner of a thick line.

A.13 INIT

Initialize BCP Library for new plot.

Format:

```
-----  
|                               |  
|          INIT (FILENAME, LUN, STATUS)  
|          INI (STATUS)        |  
|                               |  
|-----|
```

Arguments:

FILENAME	A character variable or constant filename for the graphics file. FILENAME is a file specification as defined by the <u>RSX-11M/M-Plus MCR Operations Manual</u> , Section 2.2 or the <u>VAX/VMS Command Language User's Guide</u> , Chapter 2. Whenever the filename is specified omitting the extension, the default of .FRM is assumed. FILENAME can be specified as blank characters, where the default name BCP.PLT is assumed.
LUN	An integer*2 variable or constant that specifies a logical unit number for the graphics file. If LUN is 0, a default value of 2 is assumed.
STATUS	An integer*2 variable or array element where the INIT subroutine stores a status code describing the action taken on the CALL (or request). On normal or successful return from the subroutine, STATUS has a value of 1. Otherwise, either an error or warning condition results. Refer to Appendix C for a description of this problem.

Description:

The INIT routine has two entry points, INIT and INI. The INIT entry reinitializes the BCPLIB routines for a new plot output. The user is required to call this subroutine for each plot. Forms can be defined and/or plot output generated.

The INI entry is an alternate entry point for users wishing to define forms but not to generate plot output. INI serves these users by not opening a graphics file that is never used. Phase 2 processing (PROCESS, PAGE, REPEAT) cannot be initiated following the INI call.

A.14 MOVPTR

Move current GRAPHIC POINTER position X,Y.

Format:

```
MOVPTR (X, Y, STATUS)
```

Arguments:

- X,Y Real variables or array elements specify a new position for the GRAPHIC POINTER relative to the REFERENCE ORIGIN. The units are inches and resolution is 0.1 inch.
- STATUS An integer*2 variable or array element where the MOVPTR subroutine stores a status code describing the action taken on the CALL (or request). On normal or successful return from the subroutine, STATUS has a value of 1. Otherwise, either an error or warning condition results. Refer to Appendix C for a description of this problem.

Description:

The MOVPTR subroutine moves the GRAPHIC POINTER position X inches down the page and Y inches across the page from the current REFERENCE ORIGIN.

A.15 MOVREF

Move REFERENCE pointer.

Format:

```
MOVREF (XR, YR STATUS)
```

Arguments:

- XR A real variable for the new REFERENCE X coordinate is the vertical distance in inches from the PAGE origin. XR must be at least as large as the current SECTION origin. Resolution is 0.1 inch.
- YR A real variable for the horizontal distance in inches from PAGE origin. The resolution is 0.1 inch.
- STATUS An integer*2 variable or array element where the MOVREF subroutine stores a status code describing the action taken on the CALL (or request). On normal or successful return from the subroutine, STATUS has a value of 1. Otherwise, either an error or warning condition results. Refer to Appendix C for a description of this problem.

Description:

The MOVREF subroutine moves the REFERENCE ORIGIN to a point X inches down the page and Y inches across the page from the PAGE origin or absolute position.

A.16 PAGE

Advance to the next plot page.

Terminate the current page by processing current contents of REQUEST BUFFERS and reinitialize for the next LXY page. BCP uses the LXY EVFU unit to advance the printer's paper to the next page position. This is done via the formfeed character 014 (octal).

Format:

```
-----  
|                                     |  
|                               PAGE (STATUS)                               |  
|                                     |  
-----
```

Arguments:

STATUS An integer*2 variable or array element where the PAGE subroutine stores a status code describing the action taken on the CALL (or request). On normal or successful return from the subroutine, STATUS has a value of 1. Otherwise, either an error or warning condition results. Refer to Appendix C for a description of this problem.

Description:

Page automatically calls PROCES as required to output the current REQUEST BUFFER contents.

A.17 PROCES

Initiates Phase 2 processing.

Format:

```
-----  
|                                     |  
|                               PROCES (STATUS)                               |  
|                                     |  
-----
```

Arguments:

STATUS An integer*2 variable or array element where the PROCES subroutine stores a status code describing the action taken on the CALL (or request). On normal or successful return from the subroutine, STATUS has a value of 1. Otherwise, either an error or warning condition results. Refer to Appendix C for a description of this problem.

Description:

PROCES initiates Phase 2 processing (Section 1.1) is where the PROCES subroutine is used to force the Phase 2 (Section 1.1) processing of the stored requests. The section height is determined by height of the resultant height of the graphic or plotted figures defined by the stored requests. The section height is adjusted to be a multiple of 0.5 inch. Thus, if for example the height of the graphics in the SECTION is 3.6 inches, the height is adjusted to 4.0 inches.

A.18 REPEAT

Repeat the current contents of the REQUEST BUFFERS.

Format:

```
-----  
|                                     |  
|                               REPEAT (NH, NV, STATUS)                               |  
|                                     |  
|-----|
```

Arguments:

NH	Number of times to repeat the graphics across the page (horizontally).
NV	Number of times to repeat the graphics down the page (vertically).
STATUS	An integer*2 variable or array element where the REPEAT subroutine stores a status code describing the action taken on the CALL (or request). On normal or successful return from the subroutine, STATUS has a value of 1. Otherwise, either an error or warning condition results. Refer to Appendix C for a description of this problem.

Description:

Repeat the GRAPHIC REQUESTS currently in the REQUEST BUFFERS a specified number of times. The caller specifies how many horizontal and vertical repeats are wanted.

The requests are repeated at HTAB intervals across the page and VTAB separation vertically. Automatic PAGE commands are provided upon completion of each plot page whenever the repeat count is not yet exhausted.

NOTE

It is important for the caller to get up proper values for HTAB and VTAB (see SETTAB) before calling REPEAT to set proper spacing between repeated figures.

A.19 PUTFRM

Insert a fixed form at a specified point on the plot page.

Format:

PUTFRM (X, Y, STATUS)

Arguments:

X,Y Real variables or array elements specify a new position for the REFERENCE ORIGIN for insertion of the fixed form. The units are inches and resolution is 0.1 inch.

STATUS An integer*2 variable or array element where the PUTFRM subroutines stores a status code describing the action taken on the CALL (or request). On normal or successful return from the subroutine, STATUS has a value of 1. Otherwise, either an error or warning condition results. Refer to Appendix C for a description of this problem.

Description:

PUTFRM places a fixed form at any number of places on the output PAGE. PUTFRM cannot be used when no fixed form exists (see GETFRM). One copy of the form graphic content is stored for any number of insertions by PUTFRM.

A.26 SAVFRM

Terminate form definition.

Format:

```
-----  
|                                     |  
|                               SAVFRM   (STATUS)                               |  
|                                     |  
|-----|
```

Arguments:

STATUS An integer*2 variable or array element where the SAVFRM subroutine stores a status code describing the action taken on the CALL (or request). On normal or successful return from the subroutine, STATUS has a value of 1. Otherwise, either an error or warning condition results. Refer to Appendix C for a description of this problem.

Description:

SAVFRM terminates the form definition mode of operation and writes the current contents of the REQUEST BUFFERS onto the forms file. These REQUESTS thus become the form graphics (graphic description) that may be read back and inserted in the REQUEST BUFFERS by the GETFRM subroutine at a later time. The REQUEST BUFFERS are then cleared.

A.21 SETFIG

Set up default figure parameters.

These values are used when figures such as barcode and block characters are requested.

Format:

```
-----  
|                               |  
| SETFIG (SIZE, RATIO, DBHT, TEXT, STATUS) |  
|                               |  
|-----|
```

Arguments:

- | | |
|--------|--|
| SIZE | A real variable or array element specifies character size (height and width) in inches. The resolution is 0.1 inch. The initial (or default) value for size is 1.0 inch. |
| RATIO | An integer*2 variable or constant that specifies the CODE 39 barcode ratio. This is the ratio of line widths between the thick lines (or spaces) to the thin lines (or spaces) in the barcode figure itself. Ratio is either 2 or 3. The initial or default value for ratio is two. |
| DBHT | A logical*2 variable or constant has a value of .TRUE. to draw block characters at a height that is double the width. The default or initial value for DBHT is .FALSE. |
| TEXT | A logical*2 variable or constant specifies whether to include the text when the barcode figure is drawn. The text when included is inserted as the lower 0.2 inch of the barcode figure. The default (or initial) value of text is .FALSE. |
| STATUS | An integer*2 variable or array element where the SETFIG subroutine stores a status code describing the action taken on the CALL (or request). On normal or successful return from the subroutine, STATUS has a value of 1. Otherwise, either an error or warning condition results. Refer to Appendix C for a description of this problem. |

Description:

SETFIG provides the caller with the means to change the values of the system default figure parameters. SIZE and DBHT may be temporarily overridden when making a BLOCK or BCODE subroutine call by using the appropriate arguments. These parameters are more important to the CREATE program user than the BCPLIB user. The values for RATIO and TEXT may only be changed by the SETFIG subroutine.

A.22 SETPAG

Set up default page parameters.

Format:

```
-----  
|                               |  
| SETPAG (XLIM, YLIM, EVFU, STATUS) |  
|                               |  
|-----|
```

Arguments:

- XLIM** A real variable or constant specifies page length in inches for limit checks and PAGE command use. XLIM is entered into the LXY's EVFU unit to determine the distance that the paper is to advance when the PAGE (form feed) subroutine is processed. Whenever the EVFU unit is used, XLIM cannot be greater than 22.0 inches and, in addition, must be expressed to the nearest 0.5 inch increment.
- YLIM** A real variable or constant specifies the page width in inches for limit checks. The resolution is 0.1 inch. The maximum width is 13.2 inches.
- EVFU** Logical*2 flag. If EVFU is .TRUE., bypass setup and use of the LXY EVFU unit for vertical form control. EVFU value of 0 is the same as .FALSE. The default (or initial) value for EVFU is .FALSE. When EVFU is .FALSE., the LXY EVFU is programmed to have a form length equal to SETPAG's XLIM. Whenever the PAGE routine is called, the line printer's paper is advanced according to the length defined by XLIM. When EVFU is .TRUE., the paper is advanced according to the mechanical form feed length set up on the printer itself.
- STATUS** An integer*2 variable or array element where the SETPAG subroutine stores a status code describing the action taken on the CALL (or request). On normal or successful return from the subroutine, STATUS has a value of 1. Otherwise, either an error or warning condition results. Refer to Appendix C for a description of this problem.

Description:

SETPAG provides the caller access to the system default page parameters. These parameters may be changed before the first call to a graphic figure subroutine (BLOCK, BCODE, etc.).

A.23 SETTAB

Set up default TAB distance.

Format:

```
-----  
|                               |  
|          SETTAB (HTAB, VTAB, STATUS)          |  
|                               |  
|-----|
```

Arguments:

HTAB	A real variable constant to change the value of horizontal tab length in inches. Resolution is 0.1 inch.
VTAB	A real variable or constant to change the value of vertical tab height in inches. Resolution is 0.1 inch.
STATUS	An integer*2 variable or array element where the SETTAB subroutine stores a status code describing the action taken on the CALL (or request). On normal or successful return from the subroutine, STATUS has a value of 1. Otherwise, either an error or warning condition results. Refer to Appendix C for a description of this problem.

Description:

SETTAB enables the caller to change the values for the TAB parameters (vertical and horizontal tabs). The TAB parameters are used to:

- Move the REFERENCE ORIGIN by the TAB subroutine, and/or
- Set up repeat intervals for use by the REPEAT subroutine.

✓ **

A.24 TAB

Moves REFERENCE ORIGIN incrementally. The distance is determined by VTAB and HTAB (see SETTAB).

Format:

```
-----  
|                                     |  
|                               TAB (NH, NV, STATUS)                               |  
|                                     |  
|-----|
```

Arguments:

NH	An integer*2 variable or constant specifies a number of horizontal tabs to move the REFERENCE ORIGIN.
NV	An integer*2 variable or constant specifies a number of vertical tabs to move the REFERENCE origin.
STATUS	An integer*2 variable or array element where the TAB subroutine stores a status code describing the action taken on the CALL (or request). On normal or successful return from the subroutine, STATUS has a value of 1. Otherwise, either an error or warning condition results. Refer to Appendix C for a description of this problem.

Description:

TAB enables the caller to move the REFERENCE ORIGIN multiples of predefined distances called TABS in either or both the horizontal and vertical directions.

A.25 TEXT

Draw 0.1 inch high alphanumeric characters.

Format:

```
TEXT (N, LIST, DX, DY, STATUS)
TEXTA (N, LIST, X, Y, STATUS)
TEXTP (N, LIST, P, STATUS)
```

Arguments:

- N** An integer*2 variable or array element is the number of characters in LIST (below).
- LIST** A character variable or constant containing the characters to being plotted.
- DX,DY** Real variables or array elements specify an offset distance in inches (resolution of 0.1 inch) to move the GRAPHIC POINTER for insertion of the TEXT characters.
- X,Y** Real variables or array elements specify a new position for the GRAPHIC POINTER for insertion of the TEXT characters. The units are inches and resolution is 0.1 inch.
- P** An integer*2 variable or array element specifies a form LABEL POINT. The GRAPHIC POINTER is set to the value of the named LABEL POINT for insertion of the TEXT characters. P is a number between 1 and 12.
- STATUS** An integer*2 variable or array element where the TEXT subroutine stores a status code describing the action taken on the CALL (or request). On normal or successful return from the subroutine, STATUS has a value of 1. Otherwise, either an error or warning condition results. Refer to Appendix C for a description of this problem.

Description:

TEXT allows the caller to insert text of height 0.1 inch into the plot page. The text is not affected by the SIZE and DBIT defaults that control the BLOCK character shapes.

A.26 VDASH

Dashed vertical line.

Format:

```
-----  
|  
|          VDASH (LENGTH, DX, DY, THICK, STATUS)  
|          VDASHA (LENGTH, X, Y, THICKS, STATUS)  
|          VDASHP (LENGTH, P, THICK, STATUS)  
|  
|-----
```

Arguments:

LENGTH	A real variable or constant specifies the line length in inches with a resolution of 0.1 inch.
DX, DY	Real variables or array elements specify an offset distance in inches (resolution of 0.1 inch) to move the GRAPHIC POINTER for insertion of the BLOCK characters.
X, Y	Real variables or array elements specify a new position for the GRAPHIC POINTER for insertion of the BLOCK characters. The units are inches and resolution is 0.1 inch.
P	An integer*2 variable or array element specifies a form LABEL POINT. The GRAPHIC POINTER is to be set to the value of the named LABEL POINT for insertion of the block characters. P is a number between 1 and 12.
THICK	A real variable, constant, or array element specifies the line thickness. When THICK has a value of 0, a line of normal thickness (approximately 0.039 inch) is drawn. A nonzero value of THICK specifies the actual line thickness. THICK is expressed in whole and/or tenth inch units.
STATUS	An integer*2 variable or array element where the VDASH subroutine stores a status code describing the action taken on the CALL (or request). On normal or successful return from the subroutine, STATUS has a value of 1. Otherwise, either an error or warning condition results. Refer to Appendix C for a description of this problem.

Description:

VDASH provides the caller with a facility to draw dashed vertical lines. The length of the dashed line segments and spaces between the segments is 0.1 inch.

A.27 VLINE

Dashed vertical line.

Format:

```
-----  
|                               |  
| VLINE (LENGTH, DX, DY, THICK, STATUS)  
| VLINEA (LENGTH, X, Y, THICK, STATUS)  
| VLINEP (LENGTH, P, THICK, STATUS)  
|                               |  
|-----|
```

Arguments:

LENGTH	Line length in inches with resolution of 0.1 inch (real variable).
DX,DY	Real variables or array elements specify an offset distance in inches (resolution of 0.1 inch) to move the GRAPHIC POINTER for insertion of the LINE characters.
X,Y	Real variables or array elements specify a new position for the GRAPHIC POINTER for insertion of the LINE. The units are inches and resolution is 0.1 inch.
P	An integer*2 variable or array element specifies a form LABEL POINT. The GRAPHIC POINTER is to be set to the value of the named LABEL POINT for insertion of the LINE.
THICK	A real variable or array element specifies the line thickness in inches. The resolution is 0.1 inch. When THICK has a value of 0, a line of normal thickness is drawn (approximately 0.039 inch).
STATUS	An integer*2 variable or array element where the VLINE subroutine stores a status code describing the action taken on the CALL (or request). On normal or successful return from the subroutine, STATUS has a value of 1. Otherwise, either an error or warning condition results. Refer to Appendix C for a description of this problem.

Description:

VLINE draws a solid vertical line starting at the current position of the GRAPHIC POINTER.

APPENDIX B
GLOSSARY

EVFU	LXY PRINTER/PLOTTER's electronic vertical forms unit. This unit is set up with the page length (or form feed length) and is used to advance the LXY paper to the next physical page.
FORM BASE	A point on the FORM boundary (upper left corner) that is aligned to the REFERENCE ORIGIN at the time of form insertion (see GETFRM, Appendix A, Section A.9). All FORM GRAPHICS and LABEL POINTS are stored on the FORMS file with the position (X and Y) expressed relative to the FORM BASE on the form.
FORMS	A collection of GRAPHIC COMMANDS or REQUESTS that can be saved and later called to recreate a plotted output. Additional graphic data can be inserted into the form at the time of actual output.
GRAPHIC FIGURES	Any plotted figures generated by the BCPLIB routines (such as barcodes, block character, lines, etc.).
GRAPHIC POINTER	This is insertion position on the PLOT PAGE for graphic figures (such as barcodes, block characters, lines, etc.).
GRAPHIC REQUEST	<p>This is the stored (or buffered) data that occurs when the user calls on the BCP software to draw something such as:</p> <ul style="list-style-type: none">• To the system operator (or CREATE user). This is the result of entering a graphic figure command (BCODE, BLOCK, TEXT, VLINE, HLINE, VDASH, HDASH, DEFFRM, PUTFRM) or,• To the system programmer (or BCPLIB user). This is a result of calling one of the GRAPHIC SUBROUTINES.
LABEL POINTS	A set of offset points (defined at form creation time) defined relative to the FORM BASE. LABEL POINTS can later be referenced to position variable GRAPHIC FIGURES for insertion at the time of actual plot output.

PAGE (PLOT PAGE) The PAGE (or PLOT PAGE) is the external equivalent of the LXY PRINTER/PLOTTER page. The PAGE length and width is adjustable to match the PRINTER/PLOTTER page length and width.

REQUEST A general term for GRAPHIC REQUEST. This also includes all nongraphic figure CREATE commands and BCPLIB subroutine calls (such as MOVREF, MOVPTR, etc.).

SECTION A variable sized portion of the output page. The plot data may be output by a sequence of PAGE SECTIONS instead of whole PAGES (this is usually done when there are too many GRAPHIC REQUESTS to buffer for the entire PAGE).

APPENDIX C ERROR MESSAGES

C.1 WARNINGS

- +2 REQUEST BUFFER NEARLY FULL -- This warning message occurs when five or fewer storage packets remain in the request pool.

C.2 ERROR

- 100 GRAPHIC FILE WRITE ERROR -- This error occurs when writing plot information on the GRAPHICS FILE. The REQUESTS for this SECTION are lost and the plot must be restarted. Check for device problems (such as disk full, device offline, etc.).
- 101 CANNOT OPEN GRAPHIC FILE -- This happens while trying to initialize or open a file for graphic output and an error condition occurs. It is impossible to continue with the current plot. Check for device problems (such as disk full, device off-line, illegal UIC, etc.).
- 102 GETFRM - ERROR CLOSING FORMS FILE -- The FORMS file following the contents of the defined form may be lost. Check for device problems (such as disk full, device off-line, etc.).
- 103 ILLEGAL ARGUMENT (PARAMETER) SPECIFIED -- One of the parameters supplied by the user is illegal (such as characters in a numeric field). Reenter the request.
- 104 GETFRM - CANNOT OPEN FORMS FILE -- The program cannot open the FORM file requested by the DEFFRM request. This happens if the form was not saved (SAVFRM) at the time of definition, if the form name was misspelled, or if device problems occurred (such as off-line, write locked device, etc.).
- 105 GETFRM - FORMS FILE READ ERROR -- This error occurs while reading the FORMS file as a result of the GETFRM request. Check for device problems (such as disk full, device off-line, etc.).
- 106 DEFFRM - CANNOT CREATE FORM FILE -- A FORM File was initialized (or opened) in response to a DEFFRM request and an error occurred. Check for device problems (such as illegal device, device off-line, etc.).

- 107 DEFFRM - FORMS FILE WRITE ERROR -- Error occurred while trying to write on the FORMS file following the DEFFRM request. The contents of the currently defined form may be lost. Check for device errors (such as write-locked device, device off-line, etc.).

- 108 SAVFRM - ERROR CLOSING FORMS FILE -- This error occurs while trying to close the FORMS file following the SAVFRM request. The contents of the currently defined form may be lost. Check for device problems (such as device off-line, write-locked device, etc.).

- 109 X LIMIT FAILURE (SECTION) -- The current request (TAB, MOVREF, GETFRM, or PUTFRM) defines a reference in the previous PAGE SECTION (or PAGE). If a PAGE SECTION or PAGE was processed, any further requests to draw graphic figures or part of a figure in that PAGE SECTION or PAGE cannot be accepted.

- 110 X OR Y LIMIT FAIL - OUTSIDE PAGE BOUNDARIES -- The X or Y parameter requested was outside the page boundaries. Type SHOW PAGE to view the current default page boundaries.

- 111 FORM DEFINITION - X OR Y OUTSIDE FORM BOUNDARIES -- This error occurs only during the forms definition mode. The current REQUEST (BCODE, BLOCK, TEXT, etc.) defines a figure whose area extends beyond the limits defined by the DEFFRM REQUEST.

- 112 X OR Y LIMIT FAIL - OUTSIDE REFERENCE BOUNDARIES -- The X or Y parameter requested was outside the reference boundaries. Type SHOW POS to view the current default REFERENCE ORIGIN.

- 113 Unused.

- 114 REQUEST BUFFER FULL -- No GRAPHIC REQUESTS can be accepted until Phase 2 processing is initiated. The current REQUEST was aborted. Use the PROCES, PAGE, REPEAT, FINISH (CREATE END) REQUEST to continue.

- 115 CANNOT CHANGE FIGURE PARAMETERS WHILE PLOT IN PROGRESS -- It is illegal to make a SETFIG request once plotting starts (this is following any graphic figure request such as BCODE, BLOCK, TEXT, etc.).

- 116 PUTFRM - NO FORM FIXED IN MEMORY - A form must be fixed in memory with GETFRM/F before PUTFRM can be requested.

- 117 GETFRM - FORM ALREADY FIXED IN MEMORY -- Once a form is fixed in memory the user cannot request another form (GETFRM), fixed or not fixed. The user must request ERASE FIX before another GETFRM REQUEST can be made.
- 118 SETPAG - X LIMIT BEYOND EVFU CAPABILITY -- The LXY PRINTER/PLOTTER EVFU facility has a maximum form length of 22.0 inches (in the X direction).
- 119 ILLEGAL REQUEST SEQUENCE -- The DEFFRM request cannot be made at this time because the system is already in the FORM definition mode or because plotting is already in progress.
- 121 SAVFRM - FORM NOT BEING DEFINED -- A form must be defined (DEFFRM) before it can be saved.
- 122 CANNOT PROCESS PLOT SECTION WHILE DEFINING A FORM -- The PLOT section cannot be processed (PROCES or REPEAT request) while a form is being defined.
- 123 Unused.
- 124 FORM INSERTION - X OR Y FORM LIMIT VIOLATION -- Limit violation occurred while in the form insertion mode (GETFRM by PUTFRM requests). The area covered by the defined figure (resulting from a GRAPHIC REQUEST such as BCODE, BLOCK, etc.) extends beyond the current FORM limits. The REQUEST is aborted. To continue, either make an ENDFRM REQUEST or reposition the figure.
- 125 Y LIMIT SPECIFIED IS BEYOND LXY CAPABILITY -- The Y parameter requested is beyond the 13.2 inches the LXY is capable of handling.
- 126 X LIMIT IS NOT A MULTIPLE OF 0.5 INCH -- The X parameter requested must be a multiple of 0.5 inch. PAGE or VTAB lengths (SETPAG) must be expressed in 0.5 inch increments.
- 127 PLOT NOT INITIALIZED -- The PLOT must be initialized before any requests can be entered. Use the CREATE NEWPLT command or the BCPLIB INIT subroutine to initialize the PLOT.
- 128 PLOTTING IN PROGRESS: CANNOT RESET PAGE PARAMETERS -- You cannot make a SETPAG REQUEST once plotting starts (this is following any graphic figure request such as BCODE, BLOCK, TEXT, etc.).

- 129 PLOT IS ALREADY INITIALIZED: CANNOT BE REINITIALIZED --
A PLOT was already initialized and is currently in use.
If a new plot file is needed, the current plot file must
be terminated by the CREATE user's END command or BCPLIB
user's FINISH request. After a FINISH (END) request, a
new plot can be initialized.
- 130 Unused.
- 131 REQUEST BUFFER EMPTY, NO OUTPUT POSSIBLE -- A PROCES or
REPEAT REQUEST was made, but the REQUEST BUFFERS were
empty.
- 132 REPEAT - TAB HEIGHT SMALLER THAN HEIGHT OF GRAPHICS --
The height requested with VTAB is smaller than the
height of the graphics to be repeated. The current VTAB
parameters can be viewed by typing SHOW PAG.
- 133 ILLEGAL USE OF GETFRM OR PUTFRM. MUST TERMINATE WITH
ENDFRM.
- 134 CANNOT USE "MOVREF" REQUEST DURING FORM INSERTION.
- 135 GETFRM - CANNOT FIX A FORM WHILE DEFINING A FORM --
During the form definition mode, a form cannot be fixed
in memory.
- 136 CREATE COMMAND ERROR
- 137 GETFRM - FIXED FORM DOES NOT CONTAIN GRAPHIC DATA
- 333 UNRECOGNIZED ERROR -- This can be caused by a program
"bug" or a memory error.

APPENDIX D
BCP VERIFICATION

Verification of the BCP software can be carried out on three levels.

- Verify the LXY plotting capability only
- Verify the BCPLIB graphics library
- Verify the CREATE program operation

Before starting the installation verification procedures, set the default area to that which was entered during installation. For example:

- For RSX-11M:

Set the UIC to the installation work area:

```
>SET /UIC=[200,200]
```

- For VAX/VMS:

Set the default to the installation destination work area:

```
$SET DEF DBC0:[BCP]
```

If the LXY PRINTER characteristics (lowercase, printall, width=134, and nowrap) were not set during the BCP installation, they should be set at this time. The OPER and LOG_IO privileges are needed to change the printer characteristics. The privileges and printer characteristics can be set as follows.

```
$SET PROC/PRIV=(OPER,LOG_IO)  
$SET PRINTER ddnn:/LOWERCASE  
$SET PRINTER ddnn:/PRINTALL  
$SET PRINTER ddnn:/WIDTH=134  
$SET PRINTER ddnn:/NOWRAP
```

Where ddnn: is the device name and number of the LXY PRINTER/PLOTTER.

To verify the LXY printer's plot mode of operation, the BCP distribution includes a plot image file called TEST.PLT. This tests not only the printer's plot mode operation, but also the ability of the operating system and its line printer driver to pass plot data.

Transfer the file's contents to the printer as in the following example.

- For RSX-11M:

```
>PIP LP0: = TEST.PLT
```

- For VAX/VMS:

```
$PRINT/NOFEED/DEV=ddnn: TEST.PLT
```

To verify the BCPLIB graphics library, the BCP distribution includes a special user applications program for this purpose, called DEMO. DEMO is completed and task built with the BCPLIB modules via a special command file by the same name.

- For RSX-11M:

To install DEMO, type:

```
>@DEMO
```

and respond to the command procedure's dialog as in the following examples. The operator's responses are underlined. Before calling up Demo, be sure that you have already installed BCPLIB. This installation takes approximately 4 minutes.

```
>@DEMO
>* HAVE YOU COMPILED THE DEMO PROGRAM? [Y/N]: N
>* ENTER THE NAME OF THE FORTRAN-77 TASK: [S]: F77
>F77 DEMO=DEMO.FTN
>* HAVE YOU BUILT THE BCP LIBRARY? [Y/N]: Y
>* HAVE YOU BUILT THE DEMO OVERLAY DISCRIPTOR FILE (ODL)? [Y/N]: N
>* ENTER THE NAME OF THE FORTRAN-77 OTS LIBRARY: [S]: F4POTS
>TKB @DEMOLNK
>@ <EOF>
```

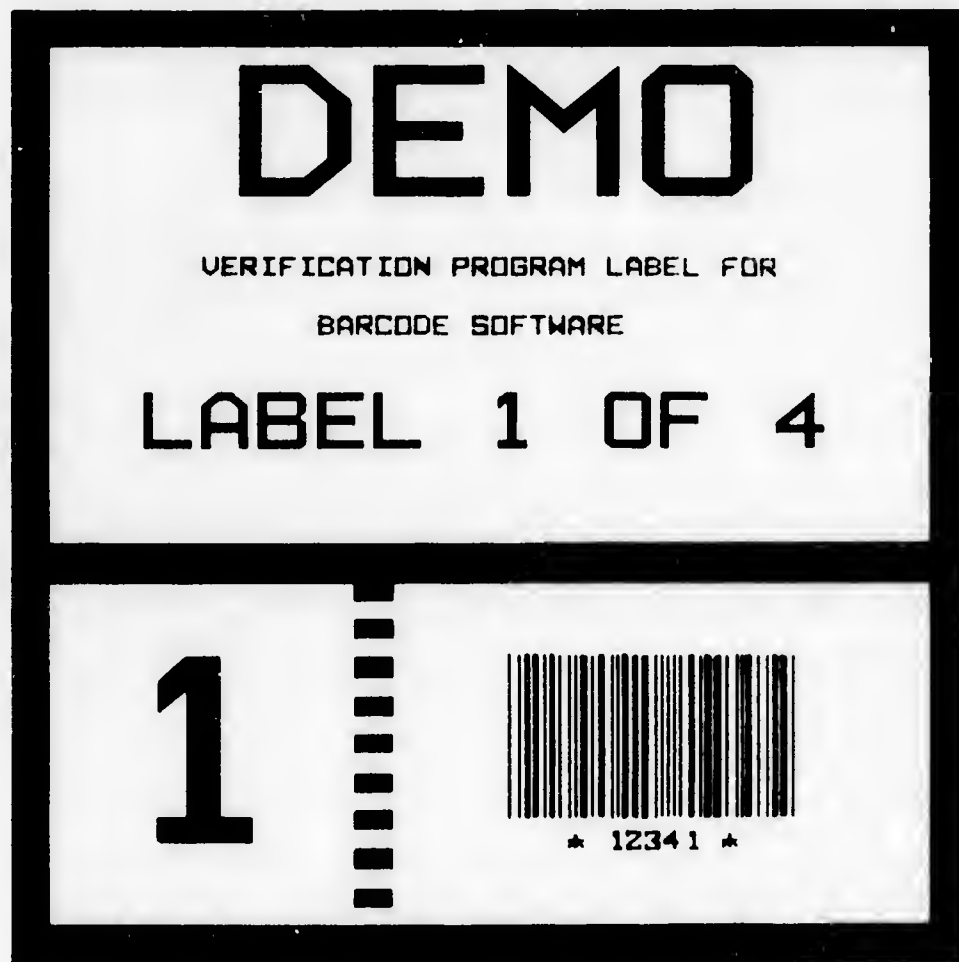
You may now run the DEMO, for example:

```
>RUN DEMO
```

Upon completion of the DEMO program, transfer the resultant graphics (plot image) file to the LXY printer.

```
>PIP LP0:=DEMO.PLT
```

Refer to Figure D-1 for the resultant plot. (Expect to see four printouts of the figure shown).



CS 2545

Figure D-1 DEMO PLOT

- For VAX/VMS:

The VAX/VMS installation procedure (Chapter 4) gives the user the option of compiling and linking the demo at the time of system installation. The demo program can be compiled and linked at any time by typing:

```
$@DEMO
```

To run the DMO program, type:

```
$RUN DEMO
```

Upon completion of the DEMO program, transfer the resultant graphics (plot image) file to the LXY printer.

```
$PRINT/NOFEED/DEV=ddnn: DEMO.PLT
```

Where ddnn: specifies a device name and number of the LXY PRINTER/PLOTTER.

Refer to Figure D-1 for the resultant plot. (Expect to see four printouts of the figure shown.)

CREATE PROGRAM VERIFICATION INSTRUCTIONS

To verify that the CREATE portion of the BCP software was properly installed and is working correctly, follow the procedure outlined below.

- For RSX-11M:

>RUN CREATE

- For VAX/VMS:

\$RUN CREATE

The following sequence applies to both RSX-11M and VAX/VMS.

Initialize the plot file.

CRT>NEWPLT CRDEMO.PLT

Now, define the form. For the CREATE DEMO, name the form CRDEMO.FRM. The form will be a 5.0 inch x 5.0 inch square, outlined with 0.2 inch thick lines. Text, block characters of different heights, lines, and barcode will make up the body of the form.

CRT>DEFFRM CRDEMO.FRM,5,5

First, draw the outline.

```
CRT>HLINE 5/T=.2
CRT>VLINE 5/A=0,4.8/T=.2
CRT>HLINE -5/A=4.8,5/T=.2
CRT>VLINE -5/A=5,0/T=.2
```

Next, insert solid and dashed lines to act as dividers.

```
CRT>HLINE 5/A=2.8,0/T=.2
CRT>VDASH 2/A=3,1.8/T=.2
```

Now, insert the block characters, text and barcode.

```
CRT>BLOCK "DEMO"/A=1,1.2/S=.7
CRT>TEXT "VERIFICATION PROGRAM LABEL FOR"/A=1.4,1.1
CRT>TEXT "BARCODE SOFTWARE"/A=1.7,1.7
CRT>BLOCK "LABEL 1 OF 4"/A=2.3,.7/S=.3
CRT>BLOCK "1"/A=4.4,.6
CRT>BCODE "12341"/A=4.4,2.7
```

Check the entries to see that they are at the specified locations. If a mistake was made while entering the graphics commands, type "ERASE REQ" and reenter the graphics commands starting at the point where the form was outlined. If the entries were correct, save the form.

```
CRT>SAVFRM
```

Using the "GETFRM" command, place the form at reference point 1,1. Terminate forms insertion with "ENDFRM". Process the contents of the buffer with the "PROCES" command. Terminate the current plot with "END", then terminate the CREATE program with "EXIT".

```
CRT>GETFRM CRDEMO.FRM,1,1
CRT>ENDFRM
CRT>PROCES
CRT>END
CRT>EXIT
```

Print the file, CRDEMO.PLT, on the line printer.

- For RSX-11M:

```
>PIP LP0:=CRDEMO.PLT
```

- For VAX/VMS:

```
$PRINT/NOFEED/DEV=ddnn: CRDEMO.PLT
```

where ddnn: is the device name and number of the LXY PRINTER/PLOTTER.

The figure generated by the procedure above generates a plotted figure as in Figure D-1.

Your comments and suggestions will help us in our continuous effort to improve the quality and usefulness of our publications.

What is your general reaction to this manual? In your judgement is it complete, accurate, well organized, well written, etc? Is it easy to use? _____

What features are most useful? _____

What faults or errors have you found in the manual? _____

Does this manual satisfy the need you think it was intended to satisfy? _____

Does it satisfy your needs? _____ Why? _____

☐ Please send me the current copy of the *Technical Documentation Catalog* which contains information on the remainder of DIGITAL's technical documentation

Name _____	Street _____
Title _____	City _____
Company _____	State/Country _____
Department _____	Zip _____

Additional copies of this document are available from

Digital Equipment Corporation
444 Whitney Street
Northboro, MA 01532

Attention: Printing and Circulation Services (NR2/M15)
Customer Services Section

Order No. AA-N269B-TK

----- Fold Here -----

----- Do not Tear - Fold Here and Staple -----

digital



No Postage
Necessary
if Mailed in the
United States

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 33 MAYNARD, MA

POSTAGE WILL BE PAID BY ADDRESSEE

Digital Equipment Corporation
Computer Special Systems
Technical Documentation Department
55 Northeastern Blvd.
Nashua, NH 03062

